6-2020

Human-in-the-Loop Learning From Crowdsourcing and Social Media

Tong Liu
tl8313@rit.edu

HUMAN-IN-THE-LOOP LEARNING FROM CROWDSOURCING AND SOCIAL MEDIA

by

Tong Liu

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in Computing and Information Sciences

B. Thomas Golisano College of Computing and
Information Sciences

Rochester Institute of Technology
Rochester, New York
June, 2020

HUMAN-IN-THE-LOOP LEARNING FROM CROWDSOURCING AND SOCIAL MEDIA

by

Tong Liu

Committee Approval:

We, the undersigned committee members, certify that we have advised and/or supervised the candidate on the work described in this dissertation. We further certify that we have reviewed the dissertation manuscript and approve it in partial fulfillment of the requirements of the degree of Doctor of Philosophy in Computing and Information Sciences.

Dr. Christopher Homan — Date
Dissertation Advisor

Dr. Cecilia Ovesdotter Alm — Date
Dissertation Committee Member

Dr. Rui Li — Date
Dissertation Committee Member

Dr. Raymond Ptucha — Date
Dissertation Committee Member

Dr. Dan Phillips — Date
Dissertation Defense Chairperson

Certified by:

Dr. Pengcheng Shi — Date
Ph.D. Program Director, Computing and Information Sciences

HUMAN-IN-THE-LOOP LEARNING FROM CROWDSOURCING AND SOCIAL MEDIA

by

Tong Liu

Submitted to the
B. Thomas Golisano College of Computing and Information Sciences Ph.D. Program in Computing and Information Sciences
in partial fulfillment of the requirements for the
Doctor of Philosophy Degree
at the Rochester Institute of Technology

Abstract

Computational social studies using public social media data have become more and more popular because of the large amount of user-generated data available. The richness of social media data, coupled with noise and subjectivity, raise significant challenges for computationally studying social issues in a feasible and scalable manner. Machine learning problems are, as a result, often subjective or ambiguous when humans are involved. That is, humans solving the same problems might come to legitimate but completely different conclusions, based on their personal experiences and beliefs. When building supervised learning models, particularly when using crowdsourced training data, multiple annotations per data item are usually reduced to a single label representing ground truth. This inevitably hides a rich source of diversity and subjectivity of opinions about the labels.

Label distribution learning associates for each data item a probability distribution over the labels for that item, thus it can preserve diversities of opinions, beliefs, etc. that conventional learning hides or ignores. We propose a humans-in-the-loop learning framework to model and study large volumes of unlabeled subjective social media data with less human effort. We study various annotation tasks given to crowdsourced annotators and methods for aggregating their contributions in a manner that preserves subjectivity and disagreement. We introduce a strategy for learning label distributions with only five-to-ten labels per item by aggregating human-annotated labels over multiple, semantically related data items. We conduct experiments using our learning framework on data related to two subjective social issues (work and employment, and suicide prevention) that touch many people worldwide. Our methods can be applied to a broad variety of problems, particularly social problems. Our experimental results suggest that specific label aggregation methods can help provide reliable representative semantics at the population level.

Acknowledgments

I am sincerely grateful to my advisor and mentor, Dr. Christopher Homan, for the invaluable guidance, motivation, support, enthusiasm, and patience he has provided throughout the years of my Ph.D. study and research. Without him, this dissertation would not have been possible.

It is also a genuine pleasure to express my gratitude to my committee members Dr. Cecilia Ovesdotter Alm, Dr. Rui Li, Dr. Raymond Ptucha, for their expertise, assistance, encouragement. I want to thank Dr. Henry Kautz, Vincent Silenzio, Qijin Cheng, Ann Marie White, and Megan Lytle-Flint to provide insights and resources to my research.

Moreover, I would like to thank Dr. Pengcheng Shi for his valuable guidance and support throughout my studies and thank Joyce hart, Lorrie Jo Turner, and Min-Hong Fu for their services.

My appreciation goes to my colleagues and good friends Akash Venkatachalam, Pratik Sanjay Bongale, Cyril Weerasooriya, James Spann, Hongda Mao, Lei Hu, Siyu Zhu, Xuan Guo, Weishi Shi, Ziyi Bai, Lingjia Deng, Cheng Guo, Mengyao Chu, Long Sha for the inspiring discussions, and their support.

My sincere gratitude also goes to my beloved parents for their unlimited and unconditional support over the years. They give me the confidence and courage to continue my study and work.

Last but not least, I would love to thank my wonderful wife and my best friend, Chen Chen, for her priceless company, support, and understanding. She brought more wisdom, courage, and joy to my life and helped me pursue my dreams and to be a better man.

To my family for their endless love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Social media are, among other things, venues for the (semi-)public disclosure of personal information that allow individuals to broadcast themselves, record events (significant or otherwise) in their lives, and express opinions and emotions whenever and wherever possible. This mixture of detail and scale in textual, temporal, and geo-spatial information that they provide is unprecedented and contributes significantly to an emerging research area, *computational social science* [15].

Due to greater activity on social media now more than ever [16], any understanding of personal or public behavioral patterns or health issues must take people's online activities into account. Personal narratives in social media can reflect an individual's experiences, states of mind, and behavioral patterns. Thus, social media is a prominent source of data on social behaviors at scales ranging from international to personal. More importantly, social media is rife with subjective domains, i.e., situations where the "best" answers to specific questions depend heavily on whom is asked, and there is no gold standard or ground truth data. For instance, taste and pain are both subjective in the sense that no authoritative or consensus-based scale exists to measure either.

Many real-world problems could have different reasonable and acceptable answers, depending on whom is asked, even when the domain of answers is fixed (i.e., *closed domain*) or more than one answer is allowed (i.e., *a multilabel setting*). For instance, events that are ordinary or unmemorable to most people can trigger intense reactions from those who have post-traumatic stress disorder (PTSD).

Subjective information is prevalent in social media: people discuss the same topic in different ways, thus greatly increasing the difficulty of computational modeling and understanding. In the

meantime, the benefits of data richness provided by social media motivate us to build computational models that are capable of extracting salient information from semi-structured, noisy social media data and generating fine-grained structured descriptions to overcome the problem of information overload [17] and facilitate downstream applications in various settings.

A typical machine learning goal is to map each given data item to a single (or set of, but in any case, deterministic) label(s) according to some standard of ground truth. However, in the cases discussed above, a single (set of) label(s) cannot meaningfully solve the problem, or may hide important dissenting beliefs or opinions. The impact of AI agents failing to recognize diversity in a representative fashion ranges from banal to harmful on a societal level. We have read several failure stories. For example, in 2016, contestants from over 100 countries from around the world submitted images of themselves to Beauty.ai's website, and their proprietary deep learning agent, trained on publicly available facial images, chose winners in 44 different beauty pageant categories [18]. The algorithm, perhaps due to biases in the trained data, showed strong signs of racial bias: 37 of the winners had distinctly European facial features [18]. Microsoft built a Twitter bot called Tay that was supposed to learn new language skills, but it had to be shut down soon after launch because it learned to deny the holocaust and demonize feminism [19]. ProPublica reported that Northpointe risk assessment software, used by judges in Florida to help determine incarceration lengths, systematically assigned higher risk scores to black defendants than to white ones [20].

Label distribution learning (LDL) replaces the conventional goal of predicting, for each data item, a single (set of) label(s) with the more challenging and complex task of predicting a probability distribution (known as a *label distribution*) over the label choices [21]. A growing body of work has used this approach, e.g., to predict beauty in images [20] and rate movies [22]. Until now, prior work has focused broadly on the problems that distinguish LDL from other forms of probabilistic learning. There is also evidence that, even in situations where ground truth exists but is difficult to obtain, predicting label distributions is more informative and accurate than aggregating the opinions of multiple labelers into a single (set of) discrete choice(s) [23]. We focus on **population-based LDL (PLDL)**, the special case of when the learning goal is to predict the distribution of beliefs in a population of human annotators about the best label(s) to associate with each data item.

A major resource bottleneck in PLDL is the number of human annotations needed to achieve a reliable learning outcome. For any large population of labelers, any individual data item x, and any multiple-choice question posed of x to the labelers, the number m of labels needed to estimate (i.e., taken as a sample of) the underlying population's true distribution of beliefs about x is rather large,

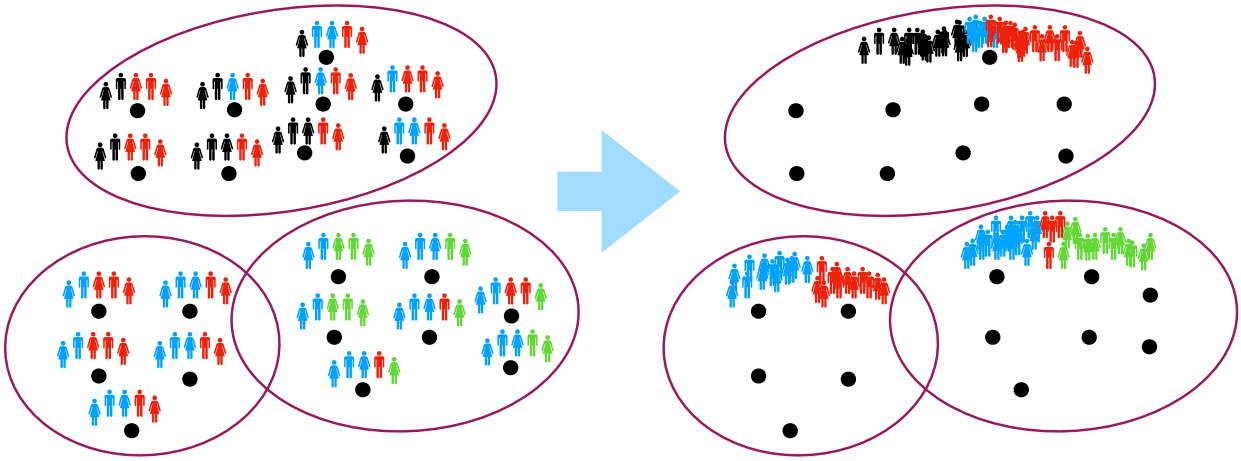

Figure 1.1: The main algorithmic idea this thesis work explores. The black dots represent data items. (Left:) Five labelers annotate each data item, where the color of the person indicates the label that person chose. If we view these five labels as a sample of the underlying population's beliefs, the sample size is probably too small for there to be much confidence in the population-level result. (Right:) We cluster together (indicated by the circles) similar rater response items, and then pool together all the labels in each cluster into a single, larger sample which, according to our learning strategy, is a good representation of—and thus label distribution for—the population-level beliefs about each item in the cluster.

depending on the size of the label space and desired confidence/significance level. Meanwhile, the number of data items n needed for supervised learning usually runs into the thousands. Thus, taken independently, the total number $m \times n$ of human labels required for training on label distributions grows quadratically and can easily run into the millions.

In this thesis work, we contribute a new algorithmic framework for reducing the total number of human labels needed per data item, by pooling together the labels of data items *determined by clustering* ***in the space of label distributions*** to be similarly rated with similar semantics. Figure 1.1 illustrates the main idea behind this approach.

1.1 Main Challenges

We face a series of social and technical challenges in developing our algorithmic solution to solve the population-based label distribution learning problems based on social media data. They are summarized as:

1.1.1 Subjectivity

Social media records individual feelings, desires, perspectives, judgments, perceptions, understandings, beliefs, and so on. All of these can be described as *subjective*, a word that is commonly used to explain various factors that have influences, impacts, or biases on people's reactions and opinions about facts or realities [24]. People with diverse life experiences may describe the same event or topic in entirely different and maybe contrary ways.

We have to deal with these subjective varieties when extracting and studying personal narratives on specific topics from social media (compared to the tasks like news and public event extractions that mainly focus on facts or truths). Subjectivity poses such a challenge to us because there is no best or correct label in subjective domains. There might be more than one acceptable solution or answer to this type of problem, or answers are judged by their acceptability rather than correctness [25].

Subjectivity comes from not only different user-generated messages themselves but also in how different readers/annotators with diverse backgrounds, perspectives, and opinions interpret these messages. To obtain high-quality annotations for further modeling and evaluation, we must deal with annotators' subjective judgments.

1.1.2 Problems without Ground Truth

When we seek to model and understand a specific social issue computationally, another obvious challenge is the lack of unambiguous and accurate definitions of the topic, i.e., there is no ground truth or gold standard available for modeling real-world problems concretely. We often have little or no authoritative datasets with which to seed, boost, validate, or evaluate our machine learning process and outcome. It proves daunting to derive the 'ground truth' underlying subjective assessments of problems such as similarities among artists' styles [26].

1.1.3 Informal Language

The language used in public social media is more informal than that of newspapers, books, forums, or blogs. Users of social media like Twitter frequently write short or incomplete phrases and use creative spellings and non-standard grammar in their tweets. Such inherent linguistic noisiness poses a significant challenge to modeling the narratives.

1.1.4 Data Scarcity

Another challenge is that any particular topic of interest may represent only a tiny fraction of total texts exchanged in any social media. Machine learning classifiers tend to perform poorly when suffering from scarcity and class-imbalance of data.

Problem Statement

Different from uncertainty, subjectivity does not assume that a data point must have a true label, even it may be just difficult or impossible to obtain. This distinction makes many probabilistic machine learning strategies inappropriate because their assumptions and designs are typically based on uncertainty measures. This distinction also has impacts on the choice of objective/loss function and evaluation metric that would be used in these modeling problems.

Our work approaches the challenges summarized above with a humans-in-the-loop label distribution learning framework, in which we develop annotation schemes to select samples for manual annotation, summarize and infer labels with probabilities from multiple annotators having various knowledge levels and opinion diversities, and develop unsupervised and supervised models to predict population-level label distributions. We collected datasets and human annotations covering different problem domains to evaluate the effectiveness and generalizability of this framework.

Figures 1.2—1.4 show the main differences between our approach and conventional supervised learning scenarios. Figure 1.2 shows the simplified relationship between a learning agent and a data item. Each item (such as a car) has observable features, and the agent's goal is to learn to predict the label of new unseen data items based on a set of labeled data. Sometimes (such as in open domain settings [27]), there is uncertainty about the labels. Figure 1.3 shows a setting where human annotators provide labels. Here, in addition to uncertainty, there may also be disagreement among the annotators as to the correct label. Nonetheless, we assume that each item has a true, gold-standard, ground-truth label. Figure 1.4 shows our setting, where there is no true hidden label (though there may be other hidden states), and the annotators' labels are a sample of the human population's interpretation of the data item. In this case, noise may be present, as before. However, the labels themselves are additional variables whose distributions depend on whom is asked, and the goal becomes to learn these underlying distributions.

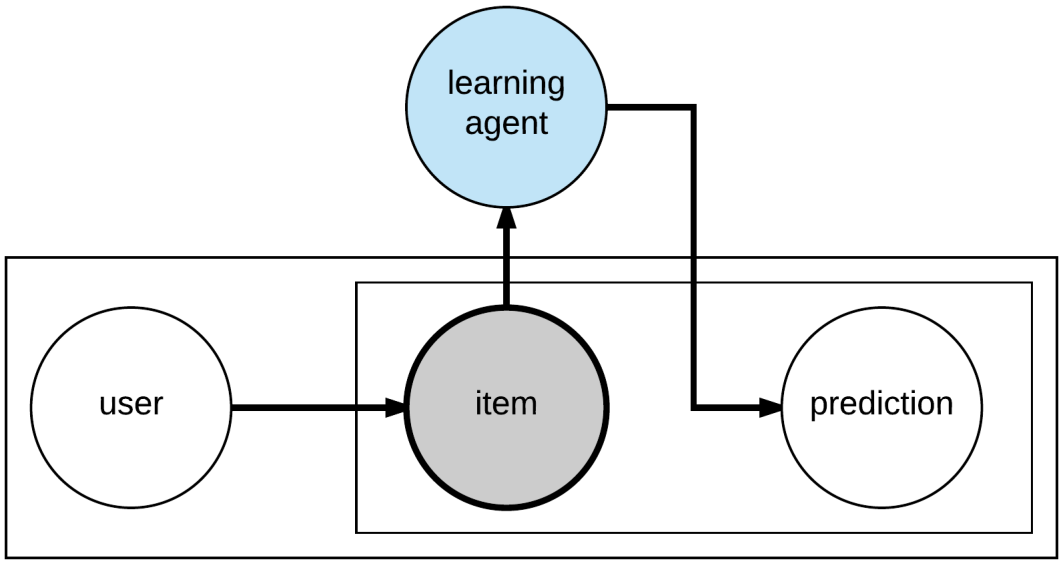

Figure 1.2: A conventional learning agent and data item. The shaded circle indicates that the data item has an observable state. The rectangles represent multivariate containments, i.e., the learning agent is dealing with multiple users, items, and predictions. Each user contributes labels to numerous items.

1.2 Thesis Contributions

This thesis makes the following contributions to human-assisted machine learning and computational social science:

1. It introduces a human-in-the-loop active learning framework that integrates the human annotations, label determinations, model training, updating and testing that progressively improve the learning performance.

2. It provides an extensible solution to solve problems like extracting complex subjective topics from massive noisy social media in a cost-effective manner. In application domains, we study the work and employment in-depth in the context of Twitter and expand to other topics.

3. It establishes the premise for our probabilistic approach through a real-world example where there is substantial disagreement over the annotators' interpretations of 50 data items in a common social domain, but where the label distributions appear visually in a histogram to

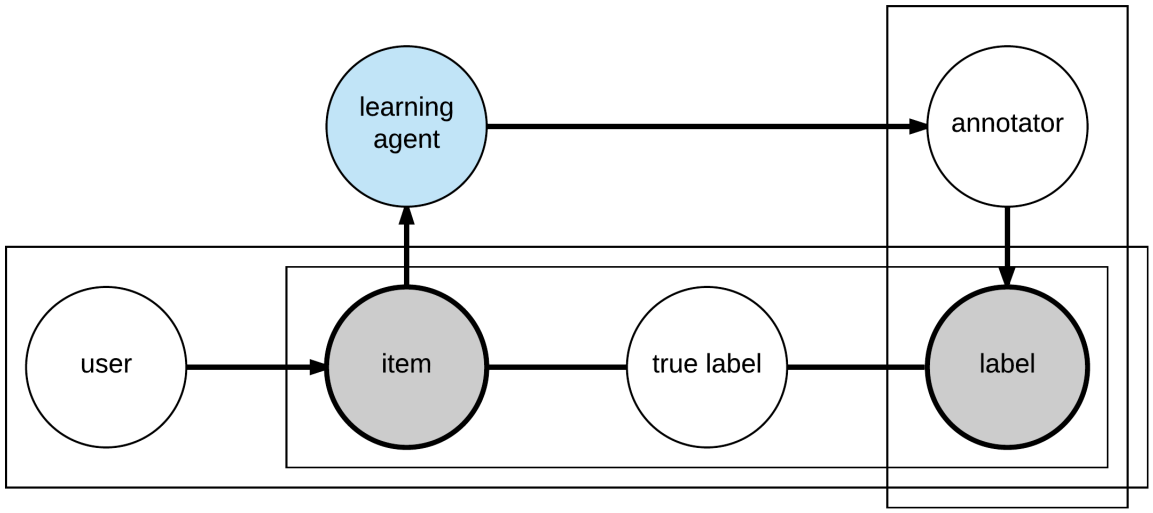

Figure 1.3: A standard learning problem. The item and label with shaded circles indicate observable conditions. Each item is assumed to have a true label, which may be different from the annotation labels.

cluster into a limited number of distinct classes.

4. It introduces an algorithmic framework for label distribution learning on as few as five-to-ten labels per data item that involves an unsupervised learning phase to yield hidden classes of semantically-related data items and assigns to each class an aggregated label distribution, followed by a supervised learning phase based on the labels the unsupervised phase produces.

5. It shows that, for larger label spaces, predictions based on unsupervised learning models that use our clustering strategy outperform those that do not, thus providing supervised learning validation for our approach.

6. Our analysis is performed on natural language data. This is among the first explorations of LDL on linguistic data from social media [28].

1.3 Thesis Outline

We organize the rest of this thesis as follows: Chapter 2 introduces the background and related work about human computation with crowdsourcing, active learning with humans in the loop,

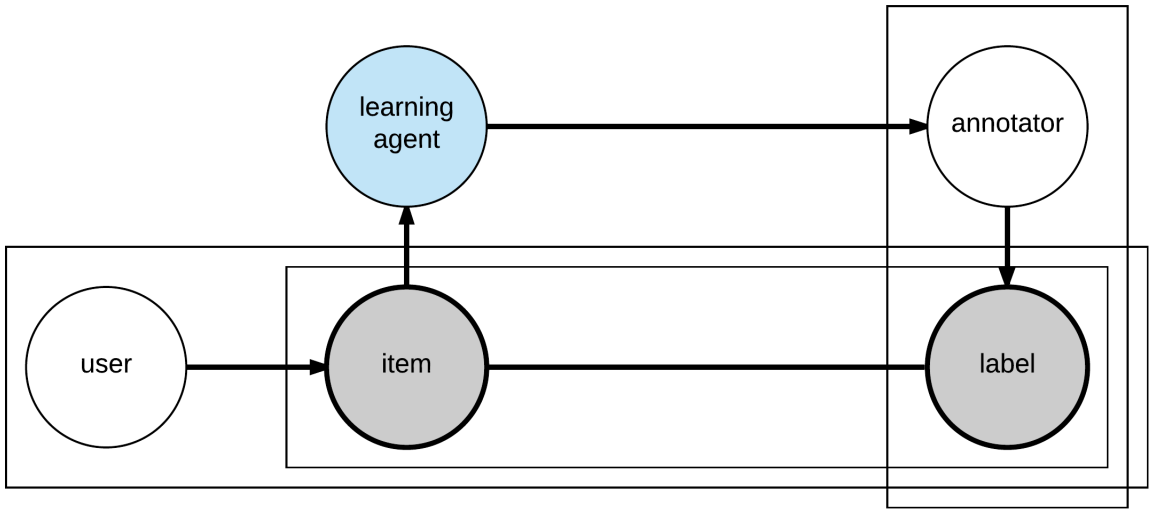

Figure 1.4: Our problem model in which there is no true label for the item. The crowdsourced labels from annotators are assumed to represent the interpretation of broad populations.

label distribution, and multilabel learning. Chapter 3 reviews our previous research about text classification of social media data built with crowdsourcing techniques. And it contains our research on comparing and combining non-expert and expert annotations in different experimental settings and introduces a humans-in-the-loop active learning framework we developed to understand social issues like work and employment. Chapter 4 introduces our label distribution learning approaches for tackling subjectivity and presents a series of experiments for evaluating our proposed learning framework. It introduces our annotation schemes for collecting labels from multiple annotators and new metrics to monitor and measure learning performance. Chapter 5 continues to discuss future work. Finally, Chapter 6 concludes our work and contributions.

Chapter 2

Background and Related Work

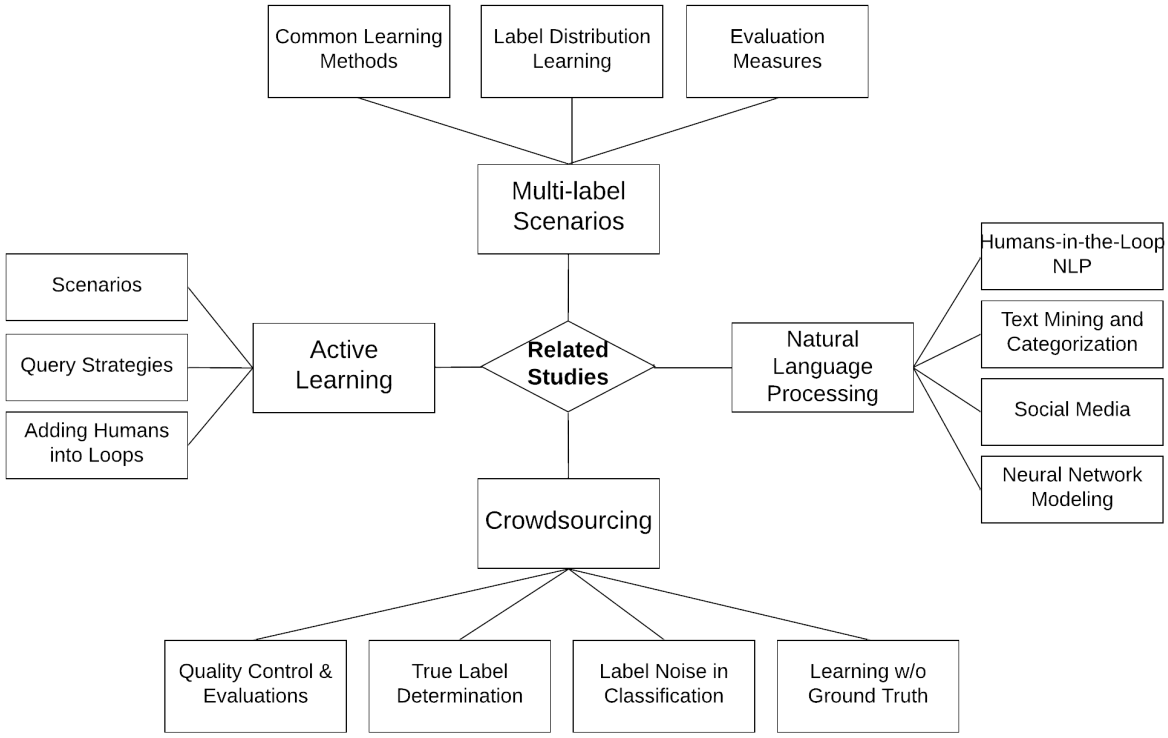

Figure 2.1: A diagram showing how different pieces of background and previous work are organized and related.

2.1 Crowdsourcing

According to Daren Brabham [29],

> Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge, and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage what the user has brought to the venture, whose form will depend on the type of activity undertaken.

Crowdsourcing is a common way for obtaining labels for supervised learning. Through a clever use of online annotation games, von Ahn pioneered the idea of crowdsourcing to collect annotations of images [30] and word relations [31]. The fast growth of crowdsourcing techniques has made the collection of large numbers of human labels much easier than before: a global base of paid workers (mostly non-experts) on various web crowdsourcing platforms can be employed to complete annotation tasks in a significantly cheaper and faster way than having experts annotate at the same scale.

Amazon Mechanical Turk (AMT), as representative of a global crowdsourcing platform, has been widely used to collect annotations. AMT is an online labor market where workers get paid (usually a small amount of money) to complete given tasks from requesters. Requesters and workers (some terms called as *Turkers*) must first create an Amazon account. Requesters publish tasks to workers in the form of *human intelligence tasks* (*HITs*) that contain an arbitrary number of tasks. Requesters can specify the number of submissions from unique workers (identified by IDs) per HIT and set up payment and reward criteria. AMT allows requesters to choose particular workers that satisfy pre-defined qualifications to work on their HITs, for example: living in a specific country or having a minimum level of experience with sufficient accepted submissions in the past. Requesters decide to approve or reject submissions from workers, and Amazon handles the payment transactions. In recent years, a variety of crowdsourcing platforms with advanced features and user-friendly interfaces have launched [32].

Snow et al. [33] investigated the effectiveness and reliability of Amazon Mechanical Turk on a variety of natural language annotation tasks (affect recognition, word similarity, textual entailment recognition, event temporal ordering, and word sense disambiguation). They collected ten independent annotations for each item in the task and found that obtaining labels from multiple non-experts could improve the data quality and reach high agreements with existing gold-standard labels provided by experts. They suggested that individual labelers (including experts) tend to have a bias (which our research [5] confirmed). They also suggested that multiple annotators may contribute to diversity, thus reducing annotation bias and noise and helping produce gold-standard quality training sets (i.e, by taking for each data item the average of multiple non-experts converges on the performance of a single expert). Callison-Burch [34] and Denkowski et al. [35] extended this series of linguistic experiments to evaluate machine translation quality and confirmed the feasibility of using AMT in complex tasks. In Callison-Burch's work [34], crowdsourcing workers rank candidate choices in order of preference (i.e., *preference ranking*) and then an overall ordering of labels is computed [36]. AMT has also been widely used to obtain transcriptions of speech [37, 38], create taxonomies [39], disambiguate word senses [40, 41], and so on.

2.1.1 Quality Control and Evaluations

The annotations collected from multiple crowdworkers are inevitably noisy, because the workers may not have specific expertise in, or may not pay full attention to, their given annotation tasks due to distraction or exhaustion, or have a poor understanding of the task that they have been asked to perform. Obtaining high-quality annotations is fundamentally important, as machine learning algorithms rely on annotated data (and labels). We discuss previous research on assuring the quality of crowdsourced annotations.

For consistency and readability purpose, Table 2.1 lists the notations used in Section 2.

Snow et al. [33] proposed methods for collecting reliable annotations from multiple annotators. They recommended hiring more workers—if budget allows—to "average out" noise or bias, and to utilize the crowdsourcing platform's compensation mechanism to collect higher-quality contributions. Also, they suggested requesting annotations only from workers with high scores[1] because they are assumed to be more reliable than those with low scores.

Besides these suggestions, Snow et al. proposed to jointly model labels and workers in order to correct for the biases of non-expert annotators [33]. Suppose the data item x has a true label

[1]Some crowdsourcing platforms provide approval rates or reliability scores of workers.

Symbol	Definition
x_i	data item ($i \in N$)
N	number of data items
z_i	(hidden) true label for x_i
y_i^k	label for x_i provided by the annotator k
$\boldsymbol{y}$	set of multiple annotations
K	number of annotators
π_k	quality of label provided by the annotator k
π	overall annotators' quality/reliability, $\pi = \{\pi_k\}_{k=1}^K$
$\boldsymbol{\zeta}$	priors on z_i

Table 2.1: The notation used in Section 2.

$z \in \{A, B\}$ (we use a binary domain here for demonstration simplicity) and crowdsourced worker k annotates x as y^k ($k \leq K$). Each worker's judgment is modeled independently of the other workers, given z as:

$$P(y^1, y^2, \ldots, y^K, z) = \Big(\prod_k P(y^k|z) \Big) p(z). \tag{2.1}$$

Then, multiple annotations are integrated via Bayes rule to infer the posterior log-odds of z for x as:

$$\begin{aligned} &\log \frac{P(z = A|y^1, y^2, \ldots, y^K)}{P(z = B|y^1, y^2, \ldots, y^K)} \\ &= \sum_k \log \frac{P(y^k|z = A)}{P(y^k|z = B)} + \log \frac{P(z = A)}{P(z = B)}. \end{aligned} \tag{2.2}$$

According to a worker's performance on a gold standard set, we can estimate their response likelihoods $P(y^k|z = A)$ and $P(y^k|z = B)$. Thus, Equation 2.2 describes a weighted voting rule based on the log likelihood ratios of workers' responses and suggests an approach for examining accuracy and reliability on annotation tasks.

2.1.2 True Label Determination

It is usually assumed that there exists a true (ground truth) label for each data point that can be determined from multiple annotations using established methods.

One straightforward approach is to take only the **unanimous vote** among multiple annotators as the single ground truth label for each data item. Setting restrictions to unanimous vote can

guarantee the quality of the final label since it achieves the highest inter-annotator agreement, especially when the number of annotators is large. However, it is not always easy for multiple annotators to reach an agreement, as we observed in a multiple-choice questions setting [6], and samples that have not been unanimously annotated became unusable.

An alternative strategy is to take the **plurality vote** of multiple annotations as ground truth. However, this approach has the problem that it assumes all the annotators are equally reliable and qualified in making judgments, which is not true in practice. Simply taking the majority vote could neglect minority opinions and skew the true label due to annotators' various knowledge levels and skill sets.

There exist a few sophisticated methods studied for inferring the true label from multiple annotations for data. Dawid and Skene [1] developed a probabilistic model to learn the weight of each annotation, as illustrated in Figure 2.2, where the (hidden) true label z_i ($i \in N$) and the annotation quality π_k jointly determines the observed label y_i^k provided by the annotator k ($k \leq K$). The quality of label π_k can be learned using EM [1], and then applied to infer the "true" labels.

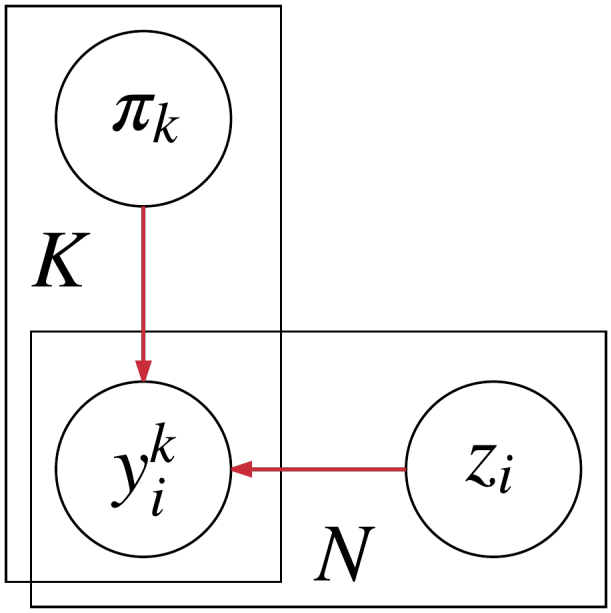

Figure 2.2: Plate notation of Dawid and Skene's model [1].

Another model for learning from multiple annotators was introduced by Welinder and Perona [2] (Figure 2.3). For a data item x_i ($i \in \{1, \ldots, N\}$), the observed label y_i^k is determined by both the

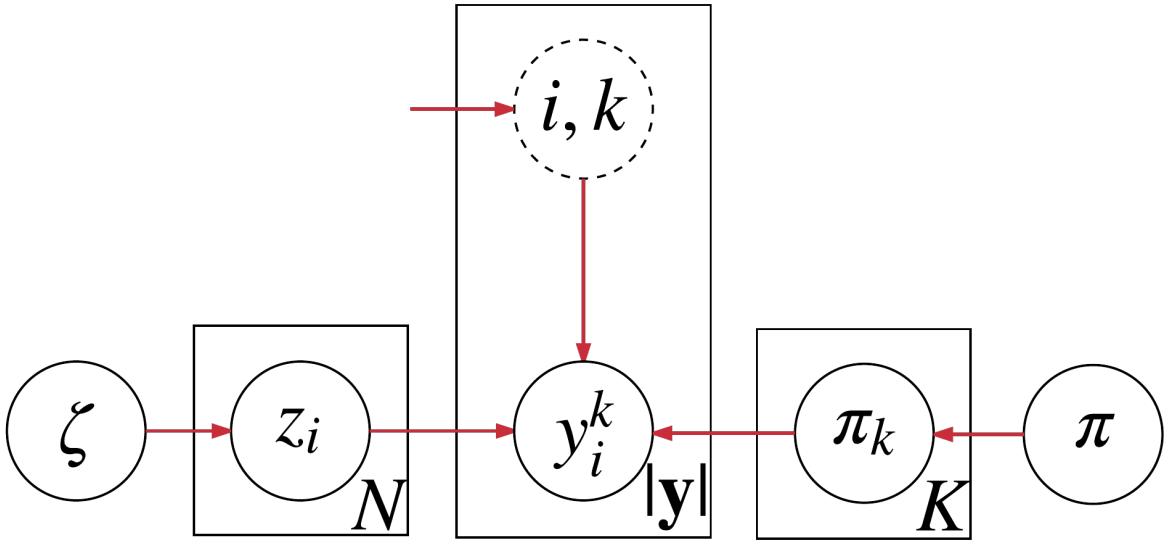

Figure 2.3: Plate notation of Welinder and Perona's model [2].

(hidden) true label z_i (parameterized by $\boldsymbol{\zeta}$) and the annotation quality π_k. The joint probability distribution of annotation process is calculated as:

$$p(y, z, \pi) = \prod_{i=1}^{N} p(z_i|\boldsymbol{\zeta}) \prod_{k=1}^{K} p(\pi_k|\pi) \prod_{y_i^k \in y} p(y_i^k|z_i, \pi_k). \tag{2.3}$$

Welinder and Perona's method can jointly estimate how reliable a particular label is and how well annotators perform. Therefore, this model can be used to validate the feasibility of our crowdsourcing annotation schemes and simultaneously evaluate the participants quantitatively.

2.1.3 Label Noise in Classification

Label noise is ubiquitous in real-world datasets and practical machine learning problems, especially those powered by multiple non-expert annotators with crowdsourcing techniques. Previous studies have various descriptions about noise: Hickey [42] defined noise as "*anything which obscures the relationship between description and class.*" Quinlan [43] defined noise as non-systematic errors. Considering the many potential negative consequences of label noise, much prior research has focused on the study of label noise and solutions to eliminate it or reduce its impact.

There are two common types of noise: (1) feature (or attribute) noise, which affects the observed

features of data items, and (2) class noise, which alters the observed labels [43, 44]. It is important to note that class noise (label noise) only affects the observed label of an instance, not its true class. Class noise is empirically shown to be more harmful than feature noise [44, 45], which can be explained by the fact that, for each instance, it is assumed there exists one observed label and multiple features. If the label is incorrectly introduced with significant class noise, the learning process would be distorted, even if the feature sets are perfectly without noise.

The observed label often corresponds to the instance's true label, but it may be subjected to noise [46, 47]. Label noise is commonly considered to be a stochastic and complex process, with independent labeling errors from each labeler [47, 48], but without a fully specified model.

Handling label noise is closely related to outlier detection [49, 50, 51, 52] and anomaly detection [53, 54, 55, 56, 57] because each mislabeled instance usually has a low chance of occurrence, and this looks anomalous to the assigned class. However, label noise does not necessarily equal the other two, such as when the labels are about subjective topics [58] or the instances themselves are simply rare events with confusing features [59].

Potential sources of label noise include: (1) the provided information being insufficient to make reliable judgments [42,60,61]; (2) unreliable annotators making mistakes in the labeling process [42]; (3) the labeling task being subjective and leading to high inter-annotator variability [62, 63, 64, 65, 66]; (4) data encoding errors and communication ambiguities leading to label noise [44, 47, 60]. Among these, the third most closely matches our problem of subjective domains.

There are three common strategies for managing label noise. The first is to employ *label noise-robust models*, which are not sensitive to and less influenced by the presence of label noise during the learning process [46]. However, this approach does not consider or handle label noise. The second strategy is to improve the quality of training data by cleaning (filtering, relabeling or removing) mislabeled instances in *label noise-polluted datasets* [46, 67]. The third approach, *label noise-tolerant learning algorithms*, takes both the classification and the label noise models into consideration simultaneously during learning [46]. Here, the third approach is relevant to our subjectivity problem, though it cannot be directly applied because it assumes there exist single true (best) labels for instances, while we do not hold such assumptions.

Under the third approach, Benoît Frénay and Michel Verleysen summarized two families of methods: probabilistic and model-based [46]. Probabilistic methods include Bayesian approaches [68, 69, 70], frequentist methods [51, 71, 72, 73, 74, 75], clustering-based methods [76, 77, 78, 79], and belief functions [80, 81, 82, 83]. Apart from probabilistic methods, many previous studies proposed label

noise-tolerant variants of popular machine learning models and algorithms, such as: SVMs with robust loss functions [84,85,86,87,88], neural networks [89,90,91,92,93], decision trees [94], boosting methods [95, 96, 97, 98, 99, 100, 101, 102], and semisupervised learning approaches [103, 104, 105, 106, 107, 108].

A number of researchers have explored **clustering** among related data items to improve the quality of inferring ground-truth labels. For instance, Zhang et al. [109] introduced a new approach, what they call ground truth inference using clustering, to better integrate labels for multi-class labeling tasks. They showed that clustering-based latent classes, compared to existing multi-class ground truth inference algorithms (majority voting, Dawid & Skene's, ZenCrowd and Spectral DS), better estimate the semantics of the data items in their comparative experiments, thus providing empirical support for a family of clustering approaches in the context of supervised learning, (which they did not study in the same work). McCallum [110] studied clustering in a semi-supervised learning context: they described a Bayesian approach to infer the distribution over mixture weights and the word distributions for documents. In both studies, clustering is performed in feature space as part of the learning process.

2.1.4 Learning without Ground Truth

Crowdsourcing facilitates the process of collecting labels from multiple annotators. Even after applying quality control and label inference approaches and canceling out label noises, there is still an underlying assumption that a (set of) correct answer(s) exists in solving these (supervised) machine learning problems. However, it is infeasible to obtain (or get access to) objective ground truth (gold standard) or indirect to confirm the ground truth for real-world problems in many scenarios.

For example, in the medical domain of cancer diagnosis, multiple radiologists may visually examine medical images (from X-ray, CT scan or MRI) and provide their subjective judgments about whether some regions are cancerous or not. One way to approach the ground truth is a biopsy of the patient's tissue, in which process it might not be cheap, safe or noninvasive. Under this circumstance, Raykar et al. [111] introduced a probabilistic supervised learning approach for training computer-aided diagnosis (CAD) classifiers without objective gold standards. They presented a solution that jointly learns a binary classifier, the annotator's reliabilities, and the actual true labels [111]. They measured annotator performance using sensitivity and specificity with respect to an unknown gold standard [111]. They presented a two-coin model (with priors to capture different skill levels) for the annotators to derive maximum-a-posterior (MAP) estimates of their reliabili-

ties. They then presented a maximum likelihood learner using the expectation maximization (EM) algorithm to iteratively estimate the posterior probabilities of ground truth labels, measure annotator performance, and update estimates of ground truth labels. Raykar et al. [111] discussed the extensibility of their algorithm to multi-class, ranking, and regression problems.

Aroyo and Welty show in a semantic parsing task that crowdworkers can perform at a level comparable to domain experts when they agree with each other. When crowdworkers have disagreement, it is often for good reason, and in fact usually more desirable than collapsing an item's annotations to a single label [112]. Schaekermann et al. [113] describe a framework for identifying unresolvable annotator disagreement. Chen et al. [114] argue persuasively that to a wide spectrum of social scientists the volume of unstructured data available for qualitative analysis generated by social media is so great that automated methods like machine learning are needed to keep up. They also argue that preserving annotator disagreement is essential to applying qualitative methodologies like grounded theory at scale.

In active learning scenarios, there is an increasing volume of research in recent years on multiple annotators (with varying expertise). Yan et al. [115] presented a probabilistic multi-labeler model in active learning loops to query the most helpful samples and, simultaneously, annotators. They modeled the observed labels for the data items by assuming two types (Gaussian and Bernoulli) of distributions based on varying reliabilities for labelers. They used a (multinomial) logistic regression model for true unknown labels in classification problems. Similar to the approach reported by Raykar et al. [111], Yan et al. [115] employed EM algorithm to calculate estimations and solved an opimization problem to achieve the joint learning goals. They applied the uncertainty sampling query strategy to select data points that have 50% accuracy in a binary scenario and calculated a template for the samples to be queried. They chose annotators whose labels had the highest confidence ratings. In a similar setting, Krempl et al. [116] presented a probabilistic active learning approach that models the true label and posterior for the query candidate. For each candidate, the expected gain in classification performance from requesting its label is calculated. Its posterior (of the positive class) is estimated based on its labeled neighborhood candidates. The candidate is then queried if its expected gain falls under the overall expected performance gain. Krempl et al. claim that their probabilistic active learning algorithm is statistically optimal in achieving density-weighted probabilistic gain in both disjoint and continuous neighborhood settings. Using their approach, querying candidates from the pool is a linear time (with respect to the size of the data pool) operation [116].

2.2 Multi-label Scenarios

When we attempt to solve subjectivity problems using machine learning, we need to account for multiple labels with probabilistic distributions for each data item. Thus, it is necessary to understand recent research in multi-label learning.

2.2.1 Common Learning Methods

Multi-label learning usually refers to either multi-label classification or multi-label ranking depending on the learning goal [117]. Multi-label classification can be described as the learning task that maps a data item $x \in \mathcal{U}$ ($\mathcal{U}$ represents the universe of data items) to more than one label (It is assumed that there exists a predefined set of labels where these multiple labels belong). Compared to the common multi-class classification problems that assume the ground truth label for each single data item is mutually exclusive, multi-label classification holds the assumption that each data item can belong to multiple classes, instead of one and only one class in multi-class classification [117]. Multi-label ranking introduces the learning task to predict the rankings of all labels, beyond the multi-label classification task that distinguishes the relevant labels from irrelevant ones [118].

Multi-label data and learning problems have attracted significant attention from academic and industrial communities and witnessed many approaches proposed and developed in recent years. Madjarov et al. [117] presented a comprehensive overview of methods for multi-label learning. They extended their previous summary [119] and summarized the methods for multi-label learning, dividing them into three categories: algorithm adaptation, problem transformation and ensemble methods [117].

The algorithm adaptation methods include adapting, extending or customizing existing machine learning algorithms for multi-label learning tasks [117], including: boosting [120, 121], k-nearest neighbors (k-NN) [122, 123, 124, 125], decision trees [126, 127], support vector machines [128], and neural networks [129].

Problem transformation methods, as the name implies, transform multi-label learning problems into single-label problems that can be solved using common machine learning algorithms [117]. Madjarov et al. grouped these problem transformation methods into three types: binary relevance, label power-set, and pair-wise methods [117]. The simplest strategy—binary relevance—converts a multi-label classification problem into a series of binary classification problems, based on the one-against-all strategy [119]. An alternative binary relevance approach is the classifier chain

method [130], which links a chain of binary classifiers. Godbole et al. [131] adapted SVM binary classifiers to multi-label learning by extending the training set and improving the class margins in forming the separating hyperplane. Label power-set methods (or label combination methods) are the second problem transformation type [119, 132, 133]. They form a single-label problem by combining the original possible label subsets into atomic (single) labels [132, 133, 134]. HOMER, a variant of label power-set methods, can learn classifiers from a hierarchical set of labels [135]. The third problem transformation type are pair-wise methods: all labels are paired with binary classifiers which can vote for the relevant labels for each data item in the multi-label learning problem [136, 137, 138, 139].

Based on the state of the art algorithm and problem transformation methods, researchers developed ensemble methods for multi-label learning. Tsomakas et al. utilized the label power-set method with a small random subset of labels to build classifiers [132]. Read et al. improved the computational efficiency of label power-set methods using the ensembles of pruned sets [133]. They also developed classifier chains methods to predict labels [130]. Kocev applied predictive clustering trees in the form of ensembles to make multi-label predictions [140].

2.2.2 Label Distribution Learning

Each data item is associated with multiple labels in multilabel learning [141]. However, it does not typically distinguish between multiplicity due to disagreement (where different annotators might believe that only one label is correct, but disagree on which one), ambiguity (where an annotator might believe multiple labels are valid), or uncertainty. Such distinctions may have significant social impacts, especially when disagreements fall along crucial demographic boundaries or indicate important but opposing perspectives that should be preserved in machine learning predictive models. Moreover, there are settings where label distributions are important but multilabel approaches do not naturally apply, such as when the prediction domain is ordinal (e.g., Likert-scaled) or real-valued. We are interested in capturing the diversity of beliefs across a population, where each member of the population may only associate a data item with one (set of) label(s), but different people may disagree on which ones.

Learning over probability distributions has a long history [1, 142, 143, 144]. While label distribution learning (LDL) adopts many of the same algorithmic approaches from this body of work it differs from conventional learning (a) in conventional probabilistic learning probability is used to model uncertainty; in LDL probabilities model ground truth. Thus (b) while conventional probabilistic learning evaluates performance in terms of accuracy, precision, and recall (even though probabilistic

measures may be used as loss functions during training) etc., in LDL performance is measured in terms of functions, such as Kullback-Leibler (KL) divergence, that operate directly on probabilities.

Geng pioneered the systematic study of label distribution learning [21], where the objects to be predicted are probability distributions over labels/classes. He and colleagues studied applications of LDL in many settings, some of which are related to predicting population-level distributions [20, 22, 23] while others are not [145, 146]. Nearly contemporary work to ours has extended the maximum entropy models in [21] to account for covariance in the label distribution space [147].

Several of these studies acknowledge the difficulty of obtaining valid label distributions that represent the underlying beliefs of human annotators; in fact, most of them are based on data and labels that were originally collected for the purpose of conventional (i.e., non-probabilistic labels) supervised learning problems. This line of research has thus far assumed that the label distributions obtained are ground truth, i.e., without questioning the statistical validity of the data, even though the sample size of the labels for each item is small.

2.2.3 Evaluation Measures

In contrast to the classical single-label learning problem, multi-label learning demands different performance evaluation measures due to the additional degrees of freedom associated with the multiple-label setting [117].

There are mainly two categories of measures for evaluating the performance of multi-label learning systems: *bipartition* and *rank-based* measures [117, 148].

For the bipartition-based measures, the differences between the predicted labels and the ground truth labels are calculated, based on either examples or labels over the test set [117]. Common example-based evaluation measures used in the multi-label experiments include hamming loss, accuracy, precision, recall, F1 score, and subset accuracy. Micro/macro-precision, micro/macro-recall and micro/macro-F1 score are common label-based evaluation measures [117].

The rank-based measures calculate the differences between the predicted ranking of multiple labels and the ground truth ranking for each example in the test set, using measures such as one-error, coverage, ranking loss, or average precision [117].

2.3 Active Learning

Subjective domains are open, without definite answers or an easy way to obtain representative labeled data. There are sometimes situations—supervised learning on social media data is one such case—where unlabeled data are abundant, but obtaining manual labels is time-consuming, expensive and difficult. In these cases, active learning algorithms can facilitate this process.

The key hypothesis behind active learning is that a machine learning algorithm can achieve better results with lower training costs if the training data are intelligently selected throughout the learning process. Active learning systems ask **queries** as an intermediate step to overcome the labeling bottleneck, and aim to achieve good performance with a limited amount of labeled data. Queries are usually in the form of unlabeled data instances that request labels from an **oracle** (such as a human annotator or another computational model/agent) [149].

Let D_0^U denote a pool of unlabeled data items with size $|D_0^U| = n$, and let $D_0^L = \emptyset$ denote the initial empty labeled set. Let D_t^L and D_t^U denote the labeled and unlabeled data respectively at time step t. The class of $x \in D_0^U$ is denoted $y \in \{1, \cdots, m\}$. Sometimes it is helpful to model the data probabilistically. In this case, x (respectively, y) is a random variable representing a data item (respectively, data label). Let $Pr^\theta(y = j|x)$ represent the probability of the sample x belonging to category j under model θ.

In terms of Mitchell's [150] definition of a well-formed machine learning problem, we define active learning for a generic classification task as:

Task function: $\boldsymbol{f} : \mathcal{U} \rightarrow \{1, \cdots, m\}$ where $\mathcal{U}$ represents the universe of data items with the corresponding true labels, and $\{1, \cdots, m\}$ represents the *classes*.

Performance metric: Precision, recall, F1-score, etc.

(Active) learning experience: The active learner selects unlabeled samples $X_t \subseteq D_t^U$ at each time step t, according to a *query strategy* S_q, and request labels for each item sampled from the *oracle*. After the query and labeling at time step t, $D_{t+1}^L = D_t^L \cup X_t$ and $D_{t+1}^U = D_t^U - X_t$.

2.3.1 Active Learning Scenarios

There are three common active learning scenarios.

Stream-based selective sampling [151, 152] (a.k.a. *stream-based/sequential active learning*) assumes that the active learner can make decisions about whether or not to request a label for an unlabeled instance when scanning through the data *sequentially* in real time to get sample at a very low cost. This approach reduces the memory size needed by the learning algorithm by either querying or discarding the current single sample (query strategies will be introduced in the next section), thus making this sampling method more appropriate for settings where memory or computational power may be limited, for example in mobile and embedded devices [149]. Several natural language processing applications have used and studied the stream-based scenario, for example in part-of-speech tagging [153], word sense disambiguation [154], information retrieval [155] and so on.

In **pool-based sampling** [156], instances to be labeled by the oracle are *greedily* and selectively queried from a large pool of unlabeled data. Compared to stream-based selective sampling, which samples the data sequentially and individually, pool-based sampling first ranks the entire unlabeled collection according to some informativeness measure before making the selection of the instances for the oracle [149].

Pool-based sampling appears to be the most popular approach and has been used in a wide range of real-world learning settings. In [149], Settles had an overview of the domains and applications where this approach has been used: text classification [156, 157, 158, 159], information extraction [160, 161], image classification and retrieval [162, 163], video classification and retrieval [164, 165], speech recognition [166], cancer diagnosis [167], etc.

There is another less common active learning scenario called **membership query synthesis** [168], in which the active learner may request labels from the oracle after creating (or synthesizing) query instances. This approach typically sends samples that are synthesized in the active learning process to the oracle, instead of using the existing instances in the data pool. This setting is usually efficient and tractable for finite problem domains [169], and can be extended to regression learning tasks like predicting robot hand coordinates in a continuous process [170]. However, this approach may in practice synthesize instances without natural meanings for the oracle, such as unidentifiable synthetic symbols in handwritten characters, or meaningless synthetic text or speech in natural language processing tasks. This makes the labeling process, especially when the oracle is a human annotator, boring and unprofitable. Stream-based selective sampling or pool-based sampling can address the above limitations.

2.3.2 Query Strategies

Settles gave a comprehensive overview of the query strategies used to evaluate and select unlabeled instances for labeling requests [149]. They are introduced below with related concepts, terms and their pros/cons.

Uncertainty Sampling

Uncertainty sampling [156] may be the most straightforward and widely used query strategy, where the active learner chooses to query the least certain samples based on some *uncertainty measure*. Measures previously used include: (1) *confidence* about the predicted label of the instances [156, 171]; (2) *margin sampling*, which incorporates the posterior of the first and second most likely labels [172]; (3) *entropy*, which considers the probability distributions over all possible class labels [173]. Compared to confidence, margin sampling could correct for the drawback that only one possible class is considered. However, margin sampling becomes a problem again when the label sets are large. Entropy-based approaches can generalize to multi-class models with more complex structured instances.

Uncertainty sampling strategies are useful for both probabilistic and non-probabilistic classification, and can also apply to regression problems (optimal experimental design) [174].

Confidence [156, 171]. All the instances in D^U are sorted according to their θ-predicted probabilities of their target class j: $Pr^\theta(y = j|x)$. Then the instances with the lowest probabilities in the sorted array are queried.

$$x_{LC} = \arg\min_x Pr^\theta(y = j|x), \tag{2.4}$$

This uncertainty measure can be interpreted as the expected 0/1-loss, i.e., the degree of lack of confidence that the model θ classifies x_i into the right class j.

Margin Sampling [172]. The confidence strategy only considers the most probable class, and ignores all other information about the remaining class distribution. For this reason, another multi-class uncertainty sampling strategy emerged [172].

$$x_{MS} = \arg\min_x (Pr^\theta(y = j_1|x) - Pr^\theta(y = j_2|x)), \tag{2.5}$$

where for the sample x, j_1 and j_2 are respectively the first and second most probable class labels predicted by the model θ. This algorithm aims to find the instance in D^U with the least margin between the two most likely classes. In other words, the least margin is a representation of the most uncertainty the model θ has towards the sample x: instances with small margins are intuitively more ambiguous to judge and predict since it is challenging for the classifier to differentiate the top two ranked classes.

Entropy [173]. Using entropy as an uncertainty measure is possibly the most general and popular uncertainty sampling strategy.

$$x_{EN} = \arg\min_{x} \sum_{j=1}^{m} Pr^{\theta}(y = j|x) \log Pr^{\theta}(y = j|x), \tag{2.6}$$

where $y = j$ ranges over all m possible categories. In information theory, entropy measures the expected value of the information encoded in a message. Here, it represents the amount of information about the label distribution: higher entropy suggests more uncertainty in the classification process. The entropy sampling approach generalizes well to probabilistic multi-class models built with complex structured instances [161].

Query-By-Committee (QBC)

The query-by-committee strategy relies on a committee of models that are all trained on the current labeled set with competing underlying hypotheses [175]. Each committee model votes on the class labels of query candidates. Then this approach selects the instances about which the committee models have the most disagreement. There are several approaches to measure the level of disagreement: (1) vote entroy [153]; (2) average Kullback-Leibler (KL) divergence [157]; (3) Jensen-Shannon divergence [176]. The QBC framework fundamentally minimizes the version space—a hierarchical representation of knowledge supplied by learning examples [150]—to precisely search for an optimal model from labeled instances, even with a small committee size [157,161,175]. This framework can be employed in regression settings [177] too.

Expected Model Change—Expected Gradient Length (EGL)

The expected model change strategy uses a decision-theoretic approach when making the instance selections. Assuming the instance label is given, we query those instances with the greatest potential

change to the current model [161, 178]. The intuition behind this strategy is that instances that could have the greatest impacts on the model's parameters are preferred. This strategy can be applied in any gradient-based learning setting. However, it is computationally expensive to apply when both the feature space and labeling sets are very large, and becomes less accurate in practice when features are not properly scaled.

Expected Error Reduction

Another approach is to measure how much the model's generalization error is likely to be reduced by the query instances [179]. After estimating the model's expected future error (total number of incorrect predictions) on the remaining unlabeled set, instances with minimal expected error are queried. This strategy can theoretically be employed to minimize loss functions and maximize any generic performance measures, such as precision, recall, F1-score, or area under the ROC curve. Compared to other query strategies, this one is the most computationally expensive and impractical in application.

Variance Reduction

This method chooses instances that minimize the model's squared-loss with respect to the objective function [180]. This strategy has advantages over approaches like error reduction in that there is no need to retrain the model since an approximated output variance simulates retraining. It has shortcomings in terms of computational complexity for larger numbers of parameters.

Density-Weighted Methods

The intuition behind the information density strategy is that the queried instances should not only be the most uncertain, but also be "representative" of the underlying distribution/density of the entire input space [161, 181]. Density-based approaches can outperform other methods [154, 157, 161, 182, 183] and allow real-time interactive active learning when densities of instances in the entire input space are pre-computed and cached for later use efficiently.

2.3.3 Adding Humans to Active Learning Loops

Humans-in-the-loop is a machine learning setting where humans serve the role of oracle.

A common assumption behind active learning is that the labels from the oracle are reliable and consistent. For example, that one can usually expect to collect labels from one single oracle, or that the oracle is always reliable in humans-in-the-loop settings. Experts who have been well trained to solve specific experience-oriented problems are often assumed to serve the roles of oracle in active learning with their professional knowledge.

Assumptions like this do not always hold in practice. Even when domain experts are the oracles, there is inevitably noise resulting in unreliable labels that can impair the performance of machine learning. Sources of noise include: (1) instances to be labeled are naturally ambiguous and subjective for people to annotate; (2) human annotators can get distracted during the annotation or become fatigued over time, especially when working on repetitive tasks; (3) their annotation skills may change (usually improve) with practice, making their submission qualities vary.

Expert annotation has other limitations: it is expensive to recruit experts to participate and the efforts of each can only produce a limited amount of labeled data. So, for some tasks that do not rely on professional knowledge, like labeling objects in images or determining the polarity of emotions in texts, one popular alternative is to use of non-experts, online or offline. With Internet-based crowdsourcing platforms, such as Amazon Mechanical Turk[2], CrowdFlower[3], non-expert annotators can be more easily and cheaply recruited to participate in annotation tasks and provide multiple labels that can be aggregated to train machine learning algorithms or evaluate machine learning models [33, 184, 185].

When either expert annotators or non-expert contributors are integrated in the active learning loops, their primary tasks, especially for subjective domains, are to provide human judgments to the queried instances. There has been much research on integrating humans in the active learning loop to study many classical problems.

Branson et al. [27] leveraged the power of both human and computer vision to recognize tightly connected objects in one picture. They provide a hybrid human-computer interaction system that progressively select and pose questions to users from a pool of predefined questions using the maximum information gain (an example of the expected model change query strategy) as

[2] `https://www.mturk.com/mturk/welcome`

[3] `https://www.crowdflower.com/`

the criterion, and produce a probabilistic output over classes for each input image. This approach achieved sufficient accuracy for practical applications as well as a significant reduction in the amount of human labor required.

Similarly, Deng et al. [186] designed a novel online game called "Bubbles" that requests users to discriminate features in heavily blurred images and classify images into given categories. Images are streamed into this process with reward mechanisms to improve performance. This feature selection challenge facilitates image categorization at sub-ordinate levels: the *BubbleBank* algorithm uses human-selected features to learn classifiers for fine-grained categories and yields large improvements over the previous state-of-the-art machine recognition benchmarks.

Gurari et al. [187] presented resource allocation methods to automatically decide, for a given batch of images and a fixed budget, when to use human annotation to create coarse segmentation samples to initialize their trained segmentation tools, and when to replace humans with computers to create high quality segmentations. This method uniquely leverages human effort at test time (instead of training time) to reduce the expected error and recover from algorithmic failures. Their results demonstrate the advantages of mixing human effort and computer algorithms in image segmentation tasks.

Moreover, there are many well-known, promising practical applications in industry that feed human judgment back into machine learning algorithms to make them perform better [188].

Google, Tesla, and major automobile companies have been developing and testing self-driving cars for years (though laws have not yet been finalized to regulate the industry [189]) and achieved state-of-the-art technologies that can drive a car safely in many situations. Such systems at this stage require human drivers to keep their hands on the wheel so they can take control back from the machine when needed to avert disasters, for example, if there is construction, a detour, snowy weather, or something unexpected on the road. Human drivers actively improve these autonomous systems by providing their driving experiences to "teach" such systems to behave more intelligently and safely.

Other examples of real-world scenarios that use humans in the loop to build and improve machine learning algorithms are photo tagging on Facebook and depositing checks in ATMs. Facebook provides automatic photo-tagging suggestions when users upload photos. Human faces are identified by face detection algorithms, and users are then prompted to confirm the predicted labels or provide their own. All of these responses are fed into the Facebook system to make it more accurate [188]. Similarly, when people deposit checks using ATMs, optical character recognition (OCR) systems

can generally understand the check amount and routing numbers to place the checks into specific account. But there are cases when handwritten numbers or language are hard to recognize, and the ATM will ask users to enter or confirm the amount using the keypad. As in the photo tagging example, human inputs provide the algorithm with more data to learn from and make it better at processing incoming tough-to-read checks [188].

Adding humans to active learning loops benefits machine learning, but also introduces noise, uncertainty, and subjectivity, which makes humans-in-the-loop learning more complicated. Subjectivity is inevitable by its nature, as people naturally have different opinions. Thus, human involvement in the learning process can lead to complications, making it more difficult to control.

2.4 Natural Language Processing

2.4.1 Humans-in-the-Loop NLP

Humans-in-the-loop active learning has been used in an increasing number of natural language processing tasks.

Ambati et al. [190] proposed a new paradigm for machine translation problems that utilizes both active learning and crowdsourcing to automatically translate language pairs. They observed significant improvements in active learning experiments when compared to existing baselines. Their crowdsourcing experiments demonstrated the feasibility of creating parallel corpora using non-expert annotators. Morgan reported a humans-in-the-loop translation process [191], where human translators in Afghanistan incrementally improve Dari translations of English medical terminology, and presented the benefits obtained from the human translators, even with small amounts of initial training data. Zaidan and Callison-Burch [192] investigated the feasibility of integrating humans-in-the-loop into a machine translation system (more specifically, minimum error rate training) and proposed a new metric for evaluating human judgment of translation quality, which led to a reusable database and reduced human feedback to the initial phase only.

2.4.2 Text Mining and Categorization

Researchers have made substantial progress in text classification tasks—which assign categories to documents—using neural network models. Kim pioneered the use of a simple convolutional

neural network model with pre-trained word vectors for sentence-level classification tasks [193]. He achieved remarkable performance in a series of experiments and proved that pre-trained word vectors are beneficial for NLP tasks using deep learning. Zhou et al. propose a unified model that combines a convolutional neural network with a long short-term memory neural network for text classification [194]. Their results show that this joint model can outperform individual CNN and RNN models by capturing both local and global features. Lee and Dernoncourt introduce a model—a combination of recurrent and convolutional neural networks—that leverages preceding information for sequential short-text classification [195]. Their model achieves state-of-the-art performance on different datasets for dialog act prediction. Rao and Spasojevic use neural network models with word embeddings and LSTM layers in different text classification experiments where the classification criteria are context-dependent specifically (actionability and political leaning) [196]. Lai et al. introduce a recurrent convolutional neural network model for text classification. The recurrent structure captures the contextual information when learning word representations and the convolutional neural network constructs the text representations [197]. They state that their method can outperform the state-of-the-art methods on different text classification datasets. Zhang et al. introduce character-level neural networks (rather than word-level models) and empirically test its effectiveness in text classification tasks by constructing a series of large scale datasets [198]. They show via comparative experiments that deep convolutional neural networks are effective and have no dependency on syntactic or semantic knowledge.

2.4.3 Social Media

Computational modeling and understanding of narratives expressed through social media data are continuously developing. The spread of infectious diseases, like flu, can be predicted through online social media [199]. Syndromic surveillance systems for multiple ailments also can be established with social media discourse [200]. Smoking and drinking abstinence, domestic abuse and other behavioral problems can be characterized and predicted [201, 202, 203]. Mental health risks and problems, such as depression and distress, and related psychological conditions have been studied in depth using social media data [10, 204, 205, 206, 207]. For example, Homan et al. [5] investigate the automatic detection of suicidal risk factors through social media, and studied different methods to collect data and annotations with various levels of expertise. Their results not only confirmed the feasibility to use social media to identify and understand suicidal risk factors and sources but also suggested the importance to keep expertise in the computational modeling loop.

There is a rich body of work exists on modeling organization behaviors, workplace affects, career

trajectories, communication and network analytics within large enterprises like HP, Microsoft, IBM [208, 209, 210, 211].

Life-changing events often precipitate physical and mental changes in health and well-being, such as distress, depression, and even suicide. Career changes are generally agreed to be one type of the most important life-changing events, as they touch most working-age adults worldwide. Our earlier investigation into distress on Twitter [5] revealed that a large number of distressful posts are related to unfavorable employment conditions and career changes. These job-related issues reflect on the productivity, well-being, quality of life etc. of individual posters. They are also associated with public health and economics at the community level. By doing so we hopefully gain insights into understanding other types of life-changing events from public social media. There is a rich body of literature focusing on major life events. Li et al. [17] demonstrated the practicability of accurate extraction of major life events (weddings, admissions, death, etc.) from Twitter based on congratulations and condolences speech acts. Choudhury et al. [204] examined patterns of online activities, emotions and language for childbirth and postnatal discourse as a case study, and observed the shifts new mothers have after child birth in their activities and emotional expressions. Choudhury et al. [212] also developed a statistical method to investigate the transitions from mental health discourse to suicidal ideation. Their approach inferred the likelihood of these shifts and derived corresponding evident markers.

2.4.4 Neural Network Models

To model social discourse in a temporal scenario where contextual information may be present, we propose to use a recurrent neural network model where connections between units can form a chain structure, which allows information to persist and exhibit dynamic temporal behavior (see Figure 2.4). This chain-like architecture reveals that recurrent neural networks have natural advantages in modeling sequential problems.

In practice, vanilla RNNs may suffer from the vanishing/exploding gradient problem in learning long-term dependencies (e.g., dependencies between far-apart steps) [213]. Thus, a variant of RNN—long short-term memory (LSTM)—was specifically designed to combat this problem [214], as Figure 2.5 shows. An LSTM unit can decide whether to keep the existing memory via three gates (input, forget, and output). When the LSTM unit detects an important feature from the input sequence at an early state, it carries this information over a long distance along the sequence and maintains such long-term dependencies, which vanilla RNNs fail to capture [4], while simultaneously forgetting unimportant features.

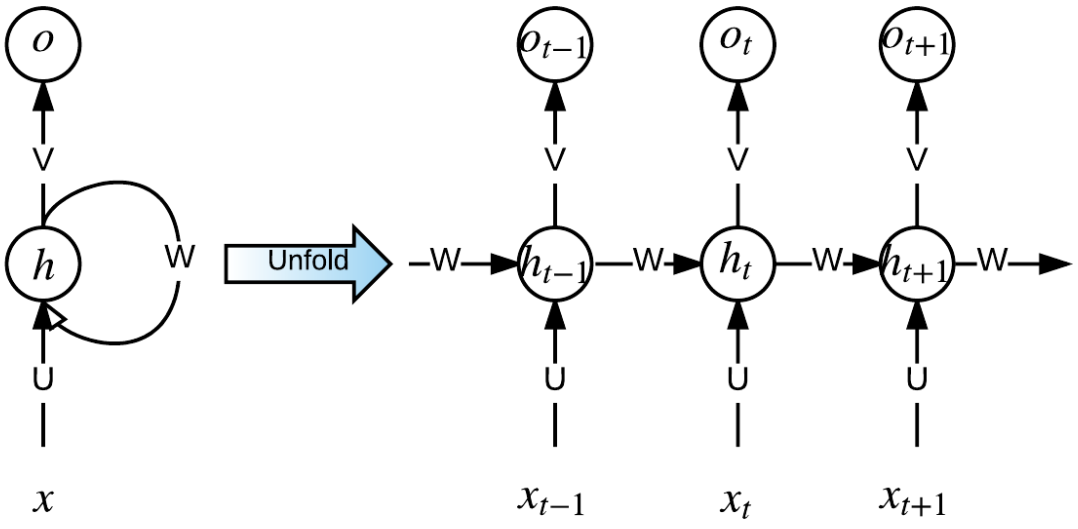

Figure 2.4: A recurrent neural network model (left) with its unfolding illustration in the process of forward computation (right). Adapted according to LeCun et al. [3].

The standard LSTM architecture consists of a series of repeated modules (illustrated in Figure 2.5), one for each time step t. At each t, three gates (the forget gate f, the input gate i, and the output gate o) collectively operate on the previous hidden state h_{t-1} and the current input x_t, and decide the update and output values and the current hidden state h_t of the current memory cell C_t, as Figure 2.4 shows.

The transition functions in a single LSTM module are defined as follows.

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ \widetilde{C_t} &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \\ C_t &= f_t \odot C_{t-1} + i_t \odot \widetilde{C_t}, \\ h_t &= o_t \odot \tanh(C_t), \end{aligned} \tag{2.7}$$

where x_t is the input vector, h_t is the output vector, $\widetilde{C_t}$ is the new memory cell candidate vector, and C_t is the new cell state. W and b are parameters of weights and bias. The symbol σ denotes

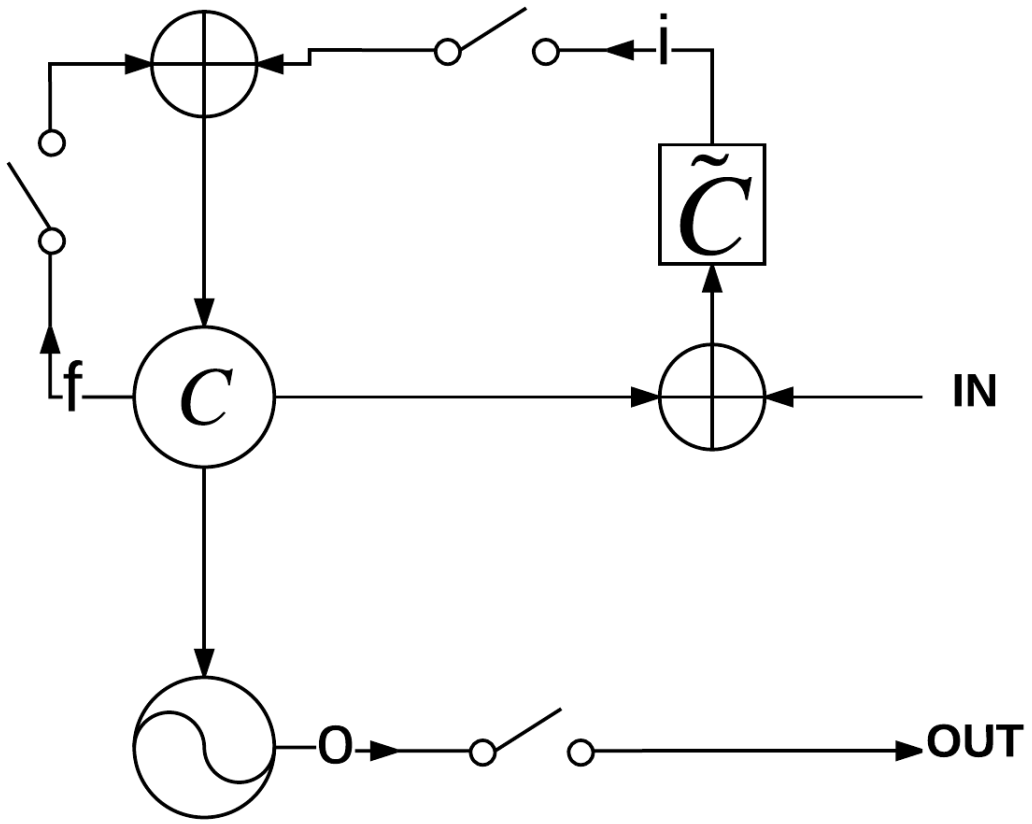

Figure 2.5: An illustration of an LSTM module. i, f and o are the input, forget and output gates, respectively. The variables c and $\tilde{c}$ demote the memory cell and the new memory cell content, respectively. Adapted from Chung et al. [4].

a logistic sigmoid function that has an output in [0, 1]. The symbol tanh denotes the hyperbolic tangent function that has an output in [-1, 1]. The symbol $\odot$ denotes element-wise multiplication. The symbols f_t, i_t and o_t denote forget gate vector, input gate vector and output gate vector, respectively. The above equations describe the gating mechanisms (input, forget, and output gates) to regulate information added or removed from the cell state: the forget gate vector f_t controls the extent the information from the old memory cell is discarded from the cell state. The input gate vector i_t decides how much the new information is going to be stored in the current memory cell. The output gate vector o_t controls what information to output (to the next cell state) based on the current memory cell. From a computational perspective, the activation function of the output

gate o_t does not depend on the memory cells state C_t, which allows us to perform part of the computation more efficiently.

LSTMs have become very popular in the field of natural language processing. They have the capability to learn to recognize context-sensitive languages with long-distance dependencies [215], which fits well the modeling of our problems.

Chapter 3

Human-in-the-Loop Active Text Mining

In this chapter, we discuss our progressive research on a humans-in-the-loop machine learning framework to model, extract, and understand mental and behavioral issues using public social media. Our framework integrates human intelligence—crowdsourced annotations and expert domain knowledge—with machine learning algorithms to gradually improve the performance of classification models.

3.1 Initial Steps to Study Social Issues

Suicide is a leading cause of death all over the world and has grown to nearly epidemic proportions in some communities [216, 217]. Suicide prevention is such a challenging problem because, relative to its social impacts, suicide is so rare and difficult to model and predict given its complexities. There is also a paucity of labeled data available to train and test predictive models. When asking people for their judgments about suicide, they have a diverse and subjective range of attitudes and beliefs about this sensitive issue. Thus, we cannot rely on simple rules or heuristics to predict suicide, or even suicidality (also known as suicide ideation). Another major challenge is that the prediction results are hard to validate due to the lack of ground truth and other difficulties (for example, ethical challenges to accessing the victims or their families). In Homan et al. [5], we took an initial step toward the automated detection of suicidal risk factors (*distress* in our case study) through social media activity with no reliance on self-reporting or interviewing. We focused on the

problem of assigning tasks and collecting labels from annotators with various degrees of expertise in suicide prevention and annotation. We compared the quality of their labels and trained multiple text-based classification models to study the effectiveness of their annotations.

We summarize our process in Figure 3.1: (1) We filter a historical Twitter dataset, obtained from Sadilek et al. [8], of approximately 2.5 million tweets from 6,237 unique users in the New York City area that were posted during a 1-month period between May and June, 2010, into a set of 2,000 tweets that are likely to be about suicide risk factors. (2) One expert and two novice annotators were instructed to label the level of distress for the selected tweets. (3) We then trained support vector machines and topic models with different combinations of annotated data and assessed the effectiveness of various annotations on the held-out set.

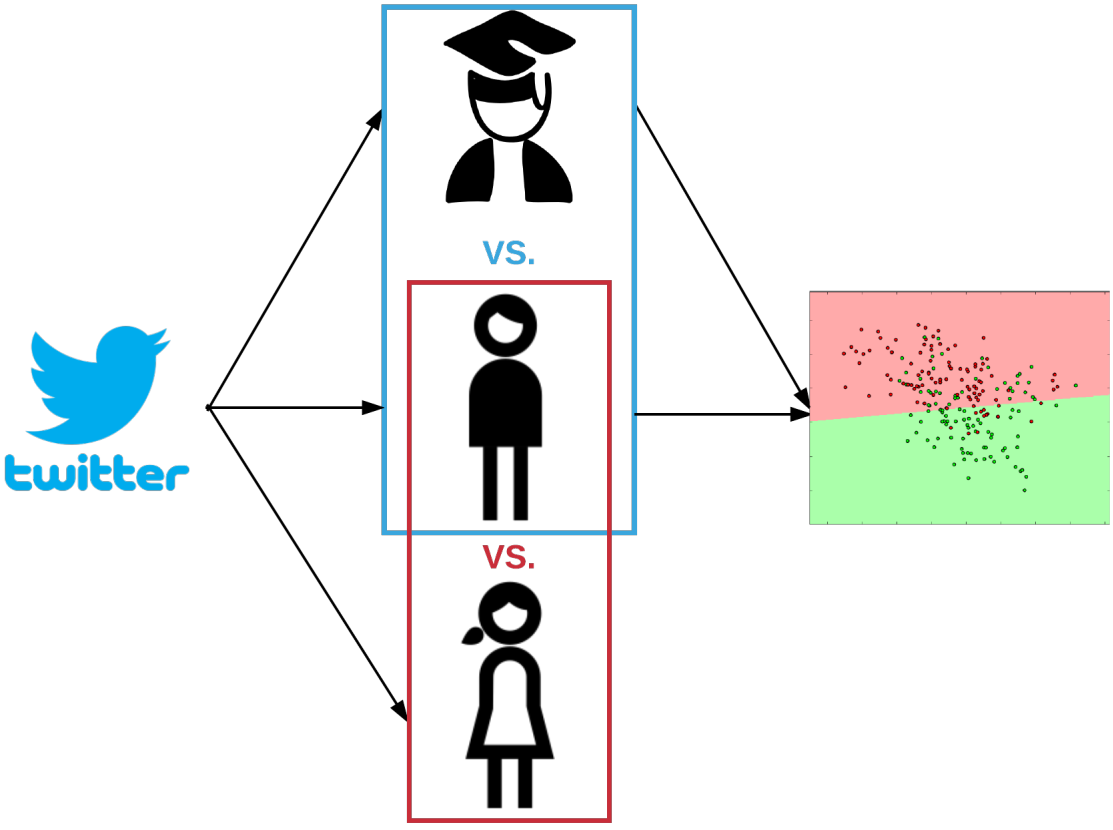

Figure 3.1: Summary of experiments in Homan et al. [5]. Twitter data are labeled by annotators with different expertise and then used to build SVM classifiers.

3.1.1 Data

Table 3.1 shows the statistics of Twitter data, and the filtered tweets using two methods as introduced below.

Source tweets	Number of tweets	2,535,706
	Unique geo-active users	6,237
	"Follows" relationships	102,739
	"Friends" relationships	31,874
Filtered tweets	Number of tweets	2,000
	Unique users	1,467
	Unique tokens	1,714,167
	Unique bigrams	9,246,715
	Unique trigrams	13,061,142
Category distribution	LIWC sad	1,370
	Depressive feeling	283
	Suicide ideation	123
	Depression symptoms	72
	Self harm	67
	Family violence/discord	47
	Bullying	10
	Gun ownership	10
	Drug abuse	6
	Impulsivity	6
	Prior suicide attempts	2
	Suicide around individual	2
	Psychological disorders	2

Table 3.1: Summary statistics and thematic category distributions of the collected dataset. The Twitter data were collected from NYC, obtained from [8]. The categories are based on LIWC [9] and Jashinsky et al. [10].

3.1.2 Annotations

Filtering Tweets

We applied two methods to filter for tweets that are likely to center on suicide risk factors and compared their effectiveness at collecting high-quality annotation data (see Table 3.1). The first method—Linguistic Inquiry and Word Count (LIWC) [9]—sampled 1,370 tweets from tweets with the 2,000th-highest *LIWC sad* scores. LIWC has been widely used to examine emotion in social networks, such as to understand mood on Twitter. The slight amount of randomness in filtering tweets avoids too many false positives being selected, for example, tweets with "sad" appeared in users' names (strings after @ symbols). Next, we adopted a collection of inclusive search terms/phrases from Jashinsky et al. [10], which was designed specifically for capturing tweets related to suicide risk factors, and applied them to our source corpus. These terms yielded 630 samples as shown in Table 3.1.

Annotation Design

In the annotation process, each tweet was provided with a context, specifically, three tweets before and after the tweet to be annotated, along with the timestamp of these tweets and thematic category to which the tweet belonged (Figure 3.2).

```
978: Date: XXXX
-3: dat man on maury is overreacting
-2: @XXXX cedes!!! [-0:21:25]
-1: yesssss! da weatherman was wronq
>>> @XXXX awwww thanks trae-trae
1: rt @XXXX: abt 2 hop in a kab to
2: @XXXX yeaa [+0:03:59]
3: @XXXX wassup? [+0:05:28]
Msg_id: XXXX  [Distress: ND, LIWC Sad: N
```

Figure 3.2: Example input for annotator. Each line is one tweet. The target tweet being annotated is indicated by >>>.

Collecting Annotations

We then divided the resulting 2,000 filtered tweets (1,370 from the LIWC *sad* dimension and 630 from suicide-specific search terms) equally into two sets (i.e., 1,000 tweets each). Both sets had the same proportion of LIWC- and suicide-specific-filtered tweets. A novice (computer science faculty) annotated the first set, and a clinical psychologist with experience in suicide-related research annotated the second. A second novice (linguistic faculty) annotated 250 tweets the first novice annotated, to reveal the inter-annotator agreement between novices, for the reason that a novice without training is expected to be less consistent. Each tweet in each set was rated on a four-point scale (H, ND, LD, HD) according to the level of distress evident (Table 3.2).

Code	Distress Level
H	happy
ND	no distress
LD	low distress
HD	high distress

Table 3.2: Distress-related categories used to annotate the tweets.

Analyzing Annotations

Figure 3.3 compares the annotation distributions in four categories between Novice 1 and the Expert. Figure 3.4 shows the distribution of annotation labels for the subset of tweets that Novices 1 and 2 mutually annotated. Table 3.3 shows the Cohen kappa score between Novices 1 and 2, when high and low distress (HD and LD) vs. no distress and happy (ND and H), are grouped into a single category, respectively (as shown in Table 3.2).

Tweets filtered by	Cohen Kappa
LIWC sad	0.4
Thematic suicide risk factors	0.6
Both	0.5

Table 3.3: Cohen's kappa inter-annotator agreement between Novice 1 and 2.

Interestingly, both novices are relatively highly conservative in assigning distressed labels, whereas the expert exhibits a higher sensitivity toward low distress (LD) than either of the novices. This suggests that it is important in this domain to not rely too much on novice judgments, as novices are not trained to pick up on subtle cues—in contrast to the clinically trained experts.

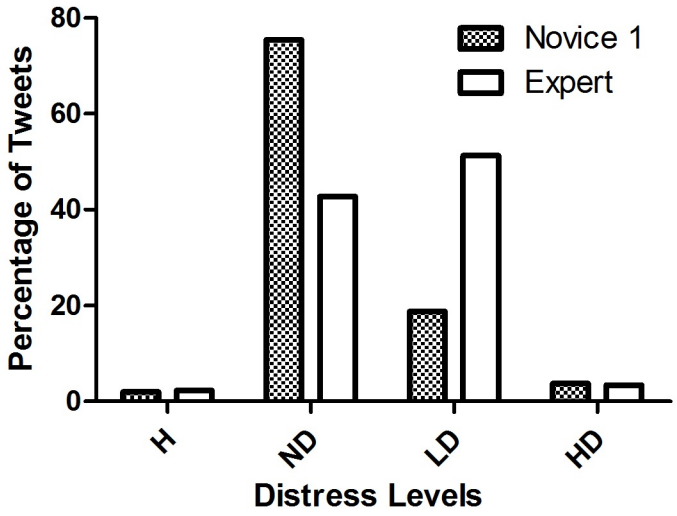

Figure 3.3: Distribution of distress level annotations from Novice 1 and Expert. Note that these two datasets are disjoint (N = 1000 tweets, respectively).

It is important to note that there are very few happy (H) tweets in Figure 3.3 and 3.4, which confirms that our filtering rules are effective.

Confusion Matrices Figure 3.4, 3.5 and 3.6 record detail numbers about annotation disagreement between Novice 1 and 2.

	H	ND	LD	HD
H	0	2	0	0
ND	1	85	2	1
LD	0	22	9	0
HD	0	1	0	2

Table 3.4: Confusion Matrix for LIWC for Novice 1 and 2.

Representative Examples

Due to their sensitive nature, we decided not to provide examples of high distress tweets. Here are two examples of tweets labeled unanimously as *low distress* by two annotators.

- *insomnia night #563258975213 65!! sheesh can't deal w/ this shit! i have class in the morning*

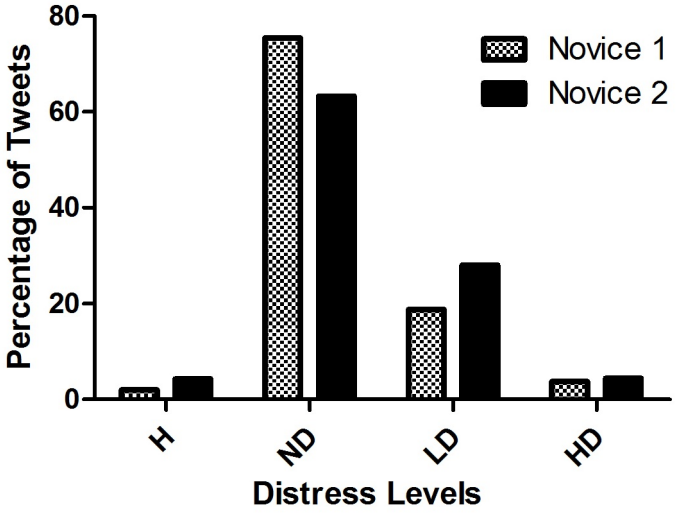

Figure 3.4: Distribution of distress level annotations on the tweets annotated by Novices 1 and 2 (N=250, identical set).

	H	ND	LD	HD
H	4	6	0	0
ND	0	55	12	1
LD	0	12	22	5
HD	0	1	3	4

Table 3.5: Confusion Matrix for Thematic Category for Novice 1 and 2.

got dammit....

- *@XXXX i'm still sad thoo. i feel neglected! and i miss XXXX*

And here are two examples of tweets labeled as *no distress* by two annotators.

- *i did mad push-ups tryna get that cut up look, then look at myself after a shower ... #plan-didntwork; thats #whyiaintgotomiami*
- *my son is gonna have blues eyes and nappy hair! yes yes yes*

Beyond the targeted annotation categories of distress level, there were emerging themes of aggression, privilege and oppression, and daily struggles, among others. For instance, jobs were a popular

	H	ND	LD	HD
H	4	8	0	0
ND	1	140	14	2
LD	0	34	31	5
HD	0	2	3	6

Table 3.6: Confusion Matrix for 250 tweets for Novice 1 and 2.

source of distress:

- *i friggin hate these bastards @ my job grimey ass bastards knew i wanted the day off and tell me some next shit*
- *as much as i hate my job some of the people i work with are amazing.*

3.1.3 Modeling Experiments

Each tweet is represented in feature space as a collection of unigrams, bigrams, and trigrams. For example, a tweet "*I am so happy*" is decomposed into {I, am, so, happy, I am, am so, so happy, I am so, am so happy}. This bag-of-words method constructs prior probabilities on pairs and triples of consecutive words and thus model the probability spaces of arbitrarily long utterances, in a way that is natural and often effective in representing textual data with contextual information (given data sparsity concerns for longer sequences) for classification, topic modeling, and so on.

We use support vector machines (SVM), a machine learning method that is used to train a classification model that can assign class labels to previously unseen tweets, to assess our collected labeled data. A support-vector machine constructs a hyperplane (or set of hyperplanes) in a high-dimensional space which can achieve the largest distance to the nearest training data points of any class [218]. SVMs treat each tweet as a point in the feature space (one dimension per uni-, bi-, and trigram in the training corpus) and act as a form of *linear separator*. They have proven to be an extremely effective tool for classifying texts in numerous settings, for different types of problems and with varying forms of social media data, including Twitter.

Because we are most interested in distinguishing distressed from non-distressed tweets, we combine low distress and high distress into one class, and no distress and happy into another (as shown in Table 3.2). Table 3.7 shows the performance of the support vector machines when trained and

tested on five combinations of the Expert and Novice 1 annotated sets. In each case, the test set is a held-out set of 100 randomly selected tweets and the remaining 900 tweets from that annotator were used as training data. The last row shows results when N1 and E data are combined into a training set of 1800 tweets and a test set of 200 tweets (with 50% of each set consisting of data annotated by Novice 1 and the Expert, respectively). Four themes emerge from the table: (1) the SVM classifier is much more accurate when the testing and training data come from one same source (the training and test sets are disjoint); (2) when testing and training data are from different sources, the SVM suffers less of an accuracy drop when the training set is from the expert than from the novice; (3) when the training set is from Novice 1, the classifier suffers a loss in recall on the distress class, and when the training set is from the Expert, there is a loss in precision instead. If our goal is to identify distress tweets, the high-precision classifier trained on Expert annotations is preferable; (4) integrating data and labels from mixed sources (i.e., novice and expert) cannot improve performance.

Training Data	**Testing Data**	**Precision**	**Recall**	**F1 Score**
N1	N1	0.53	0.63	0.58
N1	E	0.58	0.27	0.37
E	E	0.59	0.71	0.64
E	N1	0.34	0.85	0.48
N1&E	N1&E	0.33	0.41	0.37

Table 3.7: Performance of SVM-based classification when the training and testing sets are alternately Novice 1 (N1) or the Expert (E). Because we are most interested in detecting distress, we report precision and recall for the distress class, which combined LD and HD into a single **D** label in the binary classification task.

As ground truth in the classification experiments, we rely on tweets hand-annotated by an expert and a novice. However, the mental state of another individual, observed from a few lines of text often written in an informal register is essentially hard to discern and, even under less noisy conditions, extremely subjective. Also, the annotators' understandings of such concepts as "distress" may differ drastically. So a tweeter's true mental state is not revealed in an objective fashion, which makes human annotation a challenge. As we have mentioned earlier, self-reporting has its limitations, yet it is often regarded as the gold standard or the ground truth about the personal emotional state. Part of the problem in assessing the effectiveness of self-reporting is the rareness by which suicide occurs and by the inherent subjectivity of the act, which makes any data on suicide fuzzy.

3.2 Progressive Labeling in a Humans-in-the-Loop Setting

Following our initial work, we continued to focus on the topic of suicide in studying active learning techniques with humans in the loop [6], as suicide is an important but poorly understood problem, one that researchers are now seeking to better understand through social media. Due in large part to the fuzzy nature of what constitutes suicidal behavior, most supervised approaches for learning to detect suicide-related activity in social media require a great deal of human labor to train. However, humans themselves have diverse or conflicting views on what constitutes suicidal behavior. So how to obtain reliable gold standard labels is fundamentally challenging and, we hypothesize, relies heavily on what is asked of the annotators and what slice of the data they label. We conducted multiple rounds of data labeling experiments and collected annotations from crowdsourcing workers and domain experts. We aggregated the resulting labels in various ways to train a series of supervised models. Our experimental evaluations show that using unanimously agreed labels from multiple annotators is helpful to achieve robust machine models [6].

Figure 3.5 summarizes our study of progressive labeling in humans-in-the-loop framework [6]. We compared novice with expert annotators similarly to what we did previously [5]. We expand the scale of annotations to multiple annotators by progressively requesting labels from non-experts on a crowdsourcing platform and experts in suicide research field.

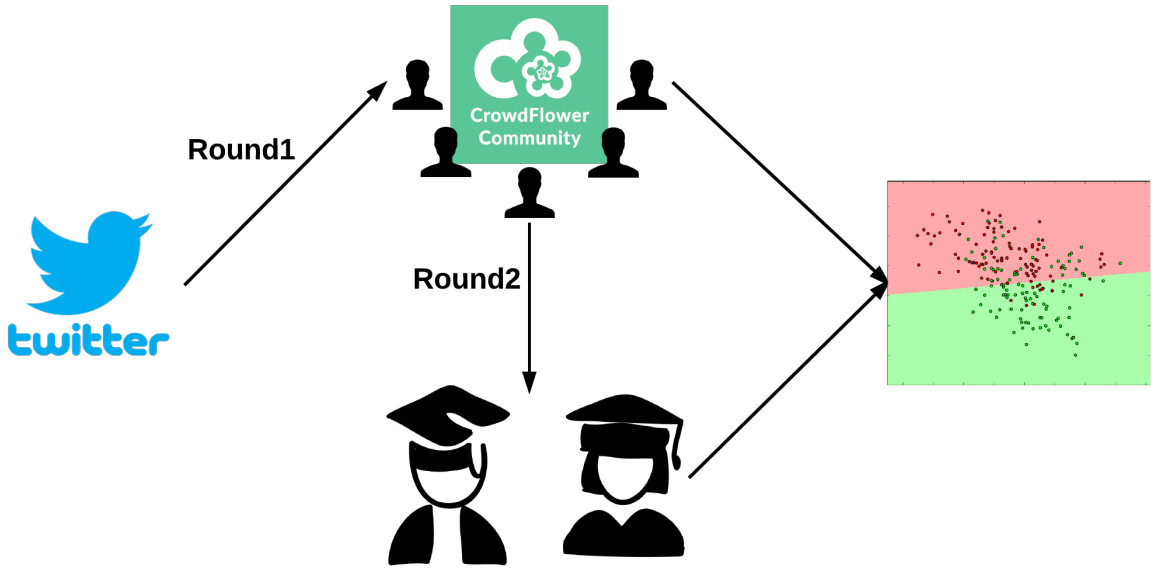

Figure 3.5: Summary of experiments in [6].

3.2.1 Data

Twitter Sampled Data – Source 1

Inspired by [207], we searched for historical Twitter posts worldwide that were related to Robin Williams's suicide case and relevant information about suicide preventions using seven key words and phrases suggested by suicide prevention experts and social workers: "*Robin Williams*", "*suicide*", "*depression*", "*Parkinson's disease*", "*seek help*", "*suicide lifeline*", and "*crisis hotline*". We downloaded via the DataSift API[1] ten percent of the Twitter messages tweeted six months before and after Robin Williams' death (August 11th, 2014) and contained at least one of the above terms. This random sampling yielded approximately 1.7 million unique tweets in English from public accounts all over the world [6].

Twitter Regional Data – Source 2

We took, as a representative sample of typical Twitter use, historical Twitter data from three metropolitan centers (Rochester, New York, and Detroit) in the United States that cover a range of population densities. Most of the tweets in this set are written in English [6].

3.2.2 Annotation Task Design

Similar to the rule-based method that we used to generate annotation materials [5], we adopted a series of pattern matching rules to obtain Twitter posts that are possibly related to suicide ideation or suicidal thoughts [219]. This rule-based filter is our initial classification model (C_0). We ran C_0 on our unlabeled dataset and randomly selected 2,000 matched tweets—1,200 tweets from *source 1* and 800 from *source 2*—for manual annotations and validations. Particularly, we anonymized the data to minimize the disclosure of personal information (@names) or URLs that may reveal clues about users' online identities to annotator.

C_0 searches for a wide range of expressions which include: *suicidal / depression / cutting / bad / sad / these ... thoughts / feelings*, *want / wanted / wanting to die*, *end / ending it all*, *end my life*, *can't take (it) anymore*, *can't / don't want to live any more*, *don't want to be alive*, *can't go on*, *call / ask for help*, *offer of help*, *stop bullying*, *kill / killing / hate myself*, *fuck / fucking*, *boyfriend*

[1] http://datasift.com/

/ girlfriend, *just ... like*, *talk / speak to someone / somebody*, *web / blog / health / advice*, *miss / missing you / her / him*, *took / taken (my / your / his / her) own life*, *hanged / hanging / overdose*, etc.

We conducted two rounds of annotations, **Round 1** (crowdsourced annotations) and **Round 2** (expert annotations), as noted in Figure 3.5. Annotators were instructed to finish a series of multiple-choice questions as detailed below.

Round 1: Crowdsourced Annotations

We first published this combination of C_0-generated 2,000 tweets on CrowdFlower[2], five tweets per page, to invite workers to finish the labeling tasks as instructed. For each tweet, five annotators were paid $1.00 to choose only one label to best describe the category from four given choices (with one sentence between the following parentheses to provide more descriptions) [6]:

- **A.** Suicidal thoughts (The author or the author's friend is at risk of suicide/distress.)
- **B.** Supportive messages or helpful information (The author is providing supportive messages/helpful information related to suicide/distress.)
- **C.** Reaction to suicide news/movie/music (The author is spreading/reacting/commenting to suicide news/movie/music.)
- **D.** Other (The author is using suicide/distress words to describe something else.)

The rationale behind the design of these multiple choice questions is: our data collection method (*source 1* especially) inevitably introduces tweets with topics covered in categories B and C among four choices. These topics are not our focus on the personal suicidal disclosure detection and so are not the primary target of our classification efforts. At the same time, this approach is useful for manually reducing the complexity of classification: Annotators can intuitively differentiate the contents of the data and establish clear boundaries between the target class (*Suicidal thoughts*) and data with related but different topics before passing them into the supervised learning algorithms [6].

[2] `https://www.crowdflower.com/`: This is an Amazon Mechanical Turk type crowdsourcing platform. Its software as a service platform allows requesters to access online workforce to clean, label and enrich data.

System Aggregated Labels (R_1S) R_1S is the majority vote of the annotators. CrowdFlower by default automatically aggregates five responses into a single result for each tweet based on the most voted label among the trusted workers[3].

Unanimous Voted Labels (R_1U) R_1U is the unanimous vote of the annotators. There are 415 tweets labeled unanimously by five workers. The remaining 1,585 tweets that were not unanimously labeled had lower inter-annotator agreements.

Round 2: Expert Annotations

Two experts were introduced to inspect tweets that crowdsourcing workers have divergent opinions. They were presented 1,585 tweets which have been labeled in Round1 and instructed to annotate according to the guidelines. For the tweets that have been unanimously agreed by crowdsourcing workers in Round1, experts do not annotate them again because crowdsourced unanimous votes are hypothesized to be as reliable as expert annotated ones.

Determining the Labels (R_2U and R_2S) The identical label from the two experts R_2U is considered to be the ground truth label. We used R_2U as the gold standard label for each tweet.

When the two experts disagree with each other on the label class, we adopted the system aggregated label in the Round1 crowdsourced annotations as the ground truth label for the data item, denoted as R_2S.

3.2.3 Annotation Summary

Table 3.8 records the percentages of tweets labeled in each of the four categories (A, B, C, and D) in five sets: they were collected respectively from non-experts and experts with distinct strategies in Round 1 and 2 (R_1S, R_1U, R_2U, R_2S) to determine the final ground truth label for each tweet and then prepared for building classification models in next phase.

We further compared the tweets in R_2U to those in R_1S and found that 871 tweets have the same ground truth label assigned by different methods (summing the diagonal numbers of the matrix in

[3]We obtained 173 tweets annotated as "A. suicidal thoughts", 265 tweets as "B. supportive messages or helpful information", 523 tweets as "C. reaction to suicide news/movie/music", and the rest 1039 as "D. other".

Category (%)	A.	B.	C.	D.	Total
R_1S	8.65	13.25	26.15	51.95	**2,000**
R_1U	2.89	8.67	17.11	71.33	**415**
R_2U	5.37	19.48	31.29	43.86	**1,042**
$R_1U + R_2U$	4.67	16.40	27.25	51.68	**1,457**
$R_1U + R_2U + R_2S$	7.85	16.00	26.55	49.60	**2,000**

Table 3.8: Statistics of labels obtained from different methods of annotations. $\boldsymbol{R_1}$: Crowdsourced annotations. $\boldsymbol{R_2}$: Expert annotations. **S**: Each tweet label comes from the system aggregated majority vote rules. **U**: Each tweet label is the unanimously voted choice among five annotators. $+$: Union operation to combine elements in different sets. **Total**: The actual counts of tweets.

Figure 3.6), which is approximately 83.59% of the total tweets with unanimous agreement between the two expert annotators.

In Figure 3.7, we compared the annotations between two experts in Round 2. Their inter-annotator agreement was assessed using Cohen's kappa [220] as $\kappa = 0.523$.

3.2.4 Modeling Experiments

Our target is to (1) identify tweets which express personal suicide ideation and suicidal thoughts and (2) differentiate between these and other types of suicide-related messages. To simply this modeling process, we grouped tweets with labels in categories B, C and D into one class and grouped the data points into binary categories: *suicidal* (positive) vs. *others* (negative) [6].

We get five sets of data items and labels in Table 3.8 into our feature extraction and modeling pipeline in order to study the influence of different labeling strategies on classification modeling performances.

Model

To control the environment variables of this experimental study, we selected support vector machines (SVMs) as our supervised learning methods to build a series of classification models. An SVM model takes in a set of training data, each labeled as belonging to one specific category, forms an optimal separating hyperplane to maximize the margin of input training data that are represented as data points in feature space. This algorithm outputs a discriminative classifier that can

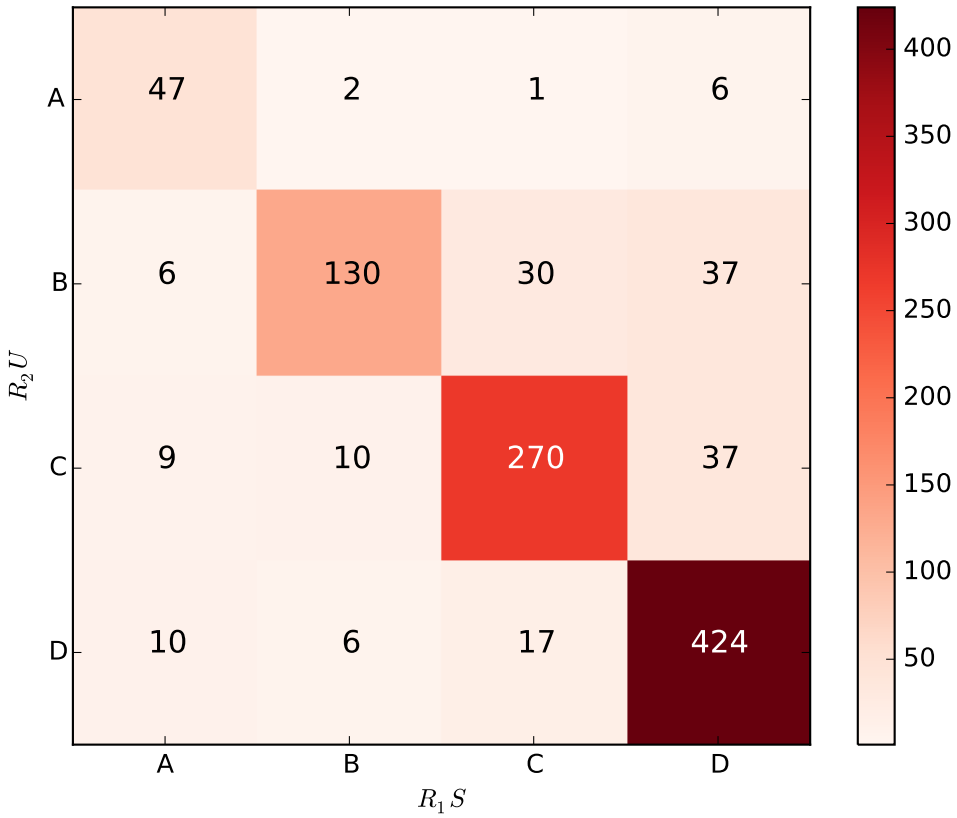

Figure 3.6: Comparisons between R_1S and R_2U.

categorize new examples (i.e., provide a predicted class label) after mapping them into the same feature space. We used the scikit-learn implementation [221] of SVMs in the experiments [6].

Feature Preparation

We used the textual representations (N-grams) to train and evaluate a series of SVM classifiers. Due to the noisy nature of Twitter, where people frequently write short or ambiguous tweets using informal spellings and grammars, we pre-processed tweets as following [6]. We: (1) replaced personal information (@names) with $@SOMEONE$, and recognizable URLs with $HTTP://LINK$, (2) utilized the `Twokenizer` system which was specially trained on Twitter texts [222] to tokenize raw messages, and (3) completed stemming and lemmatization using WordNet Lemmatizer [223].

The statistics of N-grams (unigrams, bigrams, and trigrams) extracted from different sets of training data with mixed labeling strategies are summarized in Table 3.9. We used the top **10,000** most

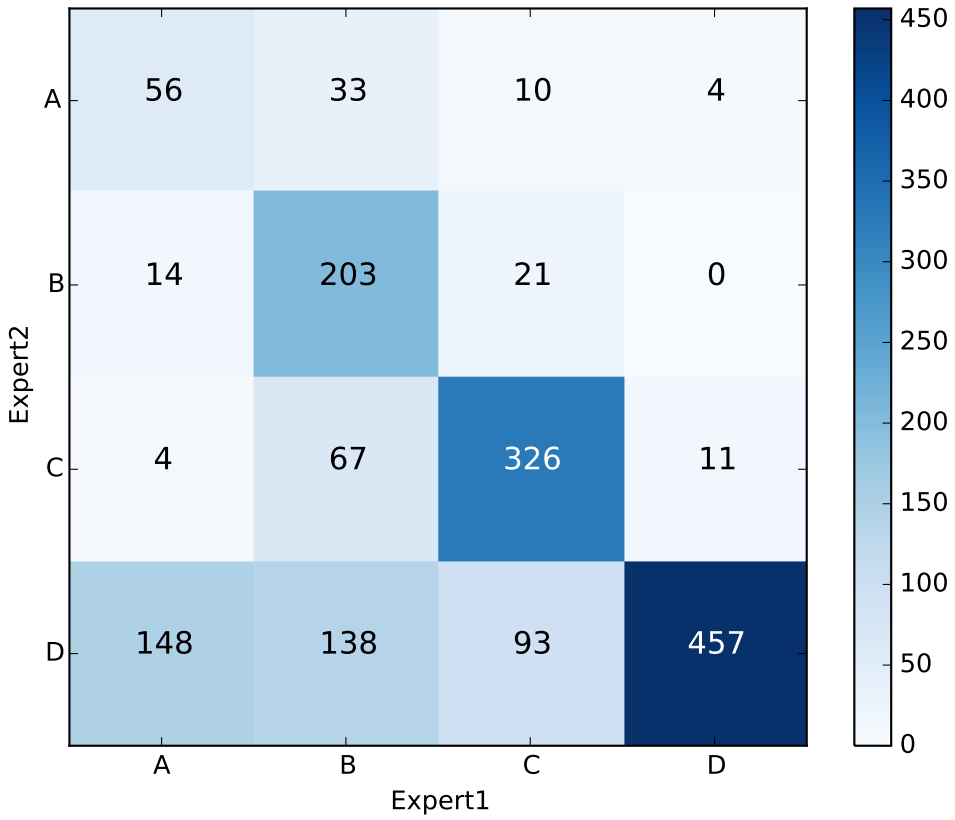

Figure 3.7: Comparisons between two expert annotators in Round 2.

frequent unique N-grams as features in the modeling process of C_1 to C_5.

Parameter Selection

Considering the class imbalance present in each training set, we determined the optimal learning parameters by grid-searching on a range of weights for the positive and negative classes, and then chose the weights that optimized the area under the receiver operating characteristic curve (AUC) [6]. We tested a variety of objective tuning functions during the grid-search process and concluded that AUC achieved the best precision, recall, and F1-score on the targeted positive class.

K-fold Cross-Validation Partitioning the available data into three sets—training, validation, and test—would drastically reduce the amount of data available for learning. The test results could depend on how the data was partitioned because of the potential over-fitting risks. We thus perform

10-fold cross-validation to evaluate our models.

3.2.5 Results and Discussions

We trained five SVM classifiers (C_1 to C_5) on five different but overlapping sets of training data, as summarized in Table 3.9.

Input Data	**Output Model**	**Uni(%)**	**Bi(%)**	**Tri(%)**	**Total N-gram Count**
R_1S	C_1	10.48	39.55	49.98	45,582
R_1U	C_2	15.89	40.75	43.36	10,493
R_2U	C_3	12.37	40.14	47.49	25,620
R_1U+R_2U	C_4	11.55	40.01	48.44	33,678
R_1U+R_2U+R_2S	C_5	10.48	39.55	49.98	45,582

Table 3.9: Statistics of features from different sources of annotations, to train models C_1 to C_5. Uni, Bi, and Tri denote unigrams, bigrams and trigrams respectively.

We analyze the similarities and differences among the five models as below.

Learning Curve

A learning curve shows the cross-validation scores of an estimator for increasing sizes of training samples, which can help us estimate how much benefit we can expect to gain by adding more training data. It also helps us understand whether the estimator suffers more from a bias error or a variance error during the modeling process[4].

Figure 3.8 shows the learning curves for models C_1 to C_5 during the training process with training data gradually added until all data are included.

We note several observations from Figure 3.8. For each model, (1) the training (dashed lines with circles) and cross-validation (solid lines with squares) scores appear to converge to each other as the size of the training set increases; and (2) at the maximum number of training samples (the ends of

[4]For an estimator, the bias error is its average error for different training sets. The variance reflects its sensitivity to varying numbers of training data points.

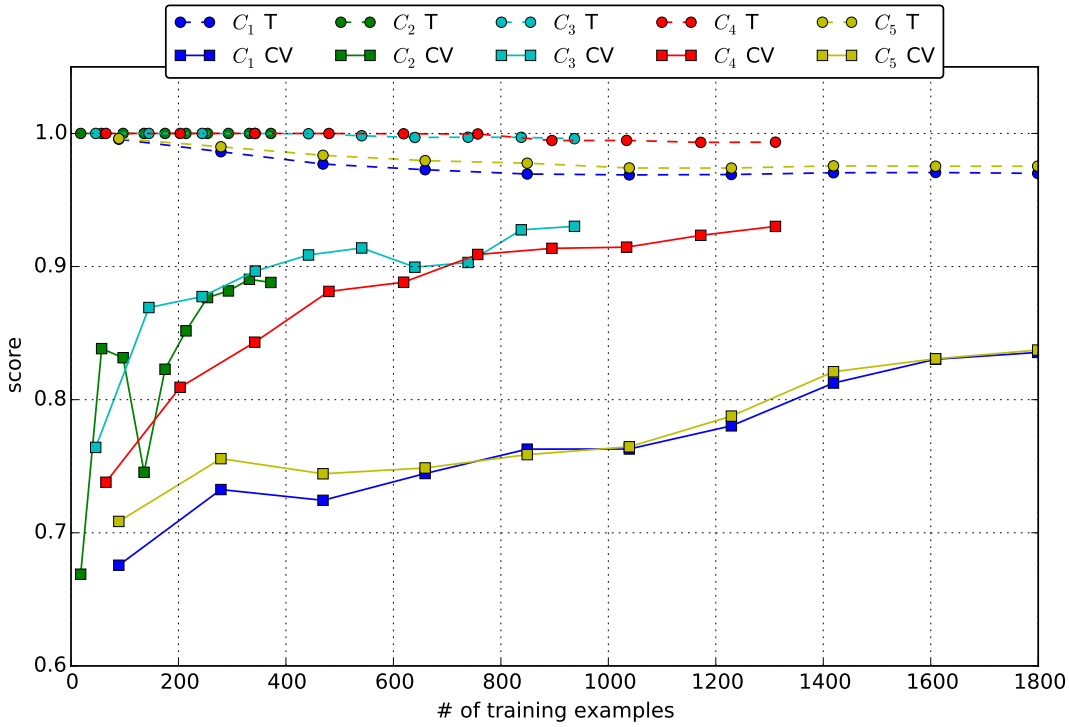

Figure 3.8: Learning curves for models C_1 to C_5 during the training process. Y axis represents area under the ROC curve. Dashed lines with circle markers represent training scores for each model, abbreviated as **T** in legend box. Solid lines with square markers represent cross-validation scores, noted as **CV**. C_1: blue; C_2: green; C_3: cyan; C_4: red; and C_5: yellow.

solid lines), the training score is much greater than the cross-validation score. These observations suggest that we can benefit from adding more training samples to improve its generalization performance as well as reduce its bias error for each model [6]. We also notice that the cross-validation scores for the five models seem to reach different points on the Y axis when the entire training set was used. Among them, C_4 has the highest cross-validation score and the least variance (according to our experiment statistics [6]), suggesting it is more stable than others. The cross-validation scores for C_2 increase and converge to the training score more quickly than the other models. However, the fluctuations of the cross-validation scores during the C_2 training process are substantial, showing its performance is not particularly stable [6].

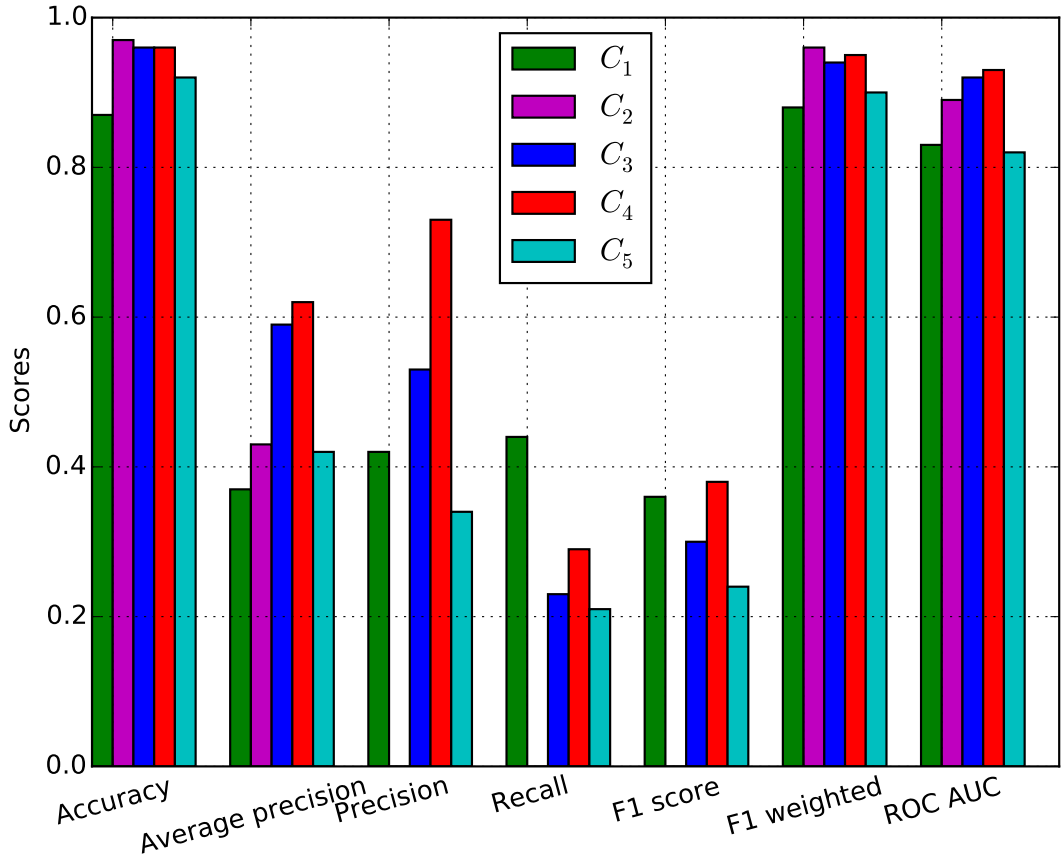

Figure 3.9: Comparisons of performance metrics for C_1 to C_5.

Performance Evaluations

Figure 3.9 compares C_1 to C_5 according to seven performance metrics. C_4 has the best *average precision*[5], *precision*, *F1* and *ROC AUC* scores. The performance of C_3 is slightly lower than that of C_4 in most measures. C_2 has score 0 in *precision*, *recall* and *F1 score* due to its bad performance—the number of correctly classified positives is 0. It is because C_2 is trained with the least amount of training data among the five, with only 12 positive samples. Even though, C_2 has performance comparable to the others in some measures, and even surpasses C_4 in *accuracy* and *F1 weighted score*[6]. This is due to the greater disparity in percentages of data items between the

[5]This score corresponds to the area under the precision-recall curve.

[6]This measure accounts for class imbalance issue. It calculates metrics for each class and finds their average, weighted by the number of true instances for each class.

positive and negative classes in R_1U (2.89% vs. 97.11%) than in the other sets of training data. C_1 and C_5 achieved lower scores than others in all measures, suggesting some relationship between the labels with lower inter-annotator agreement (R_1S and R_2S) and the robustness of predictive models.

3.3 Building a Twitter Job/Employment Corpus using the Humans-in-the-Loop Framework

In our first two suicide-related studies, jobs and employment stand out as common sources of suicide risk.

Working adults spend about one third of their lives at work. Any attempt to understand a working individual's experiences, state of mind, or motivations must take into account their life at work. Job-related social issues have been studied within enterprise-internal social platforms [208,210,211]. However, such internal services by their nature may discourage participants from fully disclosing their feelings about work, especially when work is causing them distress or they are seeking job changes.

We study Twitter as a source of job-related discourse. We constructed and developed a humans-in-the-loop supervised active learning framework that integrates crowdsourced feedback and local community knowledge to detect job-related messages from individual and business accounts. Crowdsourced validation confirms that our model can accurately identify job-related tweets. We further examined job-related discourse from an ethnographic perspective by fusing language-based analysis with temporal, geospatial, and labor statistics information.

We introduce our methodology and a dataset resulting from these methods—Twitter Job/Employment Corpus—in this section.

3.3.1 Data

Using the DataSift[7] Firehose, we collected historical tweets from public accounts with geographical coordinates located in a 15-county region surrounding Rochester, NY from July 2013 to June 2014. This one-year data set contains over 7 million geotagged tweets (approximately 90% written in

[7] `http://datasift.com/`

English) from around 85,000 unique Twitter accounts. This particular locality has geographical diversity, covering both urban and rural areas and providing mixed and balanced demographics.

3.3.2 Humans-in-the-Loop Framework

Our humans-in-the-loop supervised learning framework performs multiple iterations of learning, which integrates crowdsourcing contributions and knowledge from the local community, to perform job-related tweet identification tasks, as Figure 3.10 shows. We divided these extracted tweets into two sources: personal and business accounts, based on linguistic features [7]. Specifically, human annotators—crowdsourcing workers or community experts—actively provided feedback during the learning process to improve the feature sets. This framework served to reduce the amount of human effort needed to acquire large amounts of high-quality labeled data.

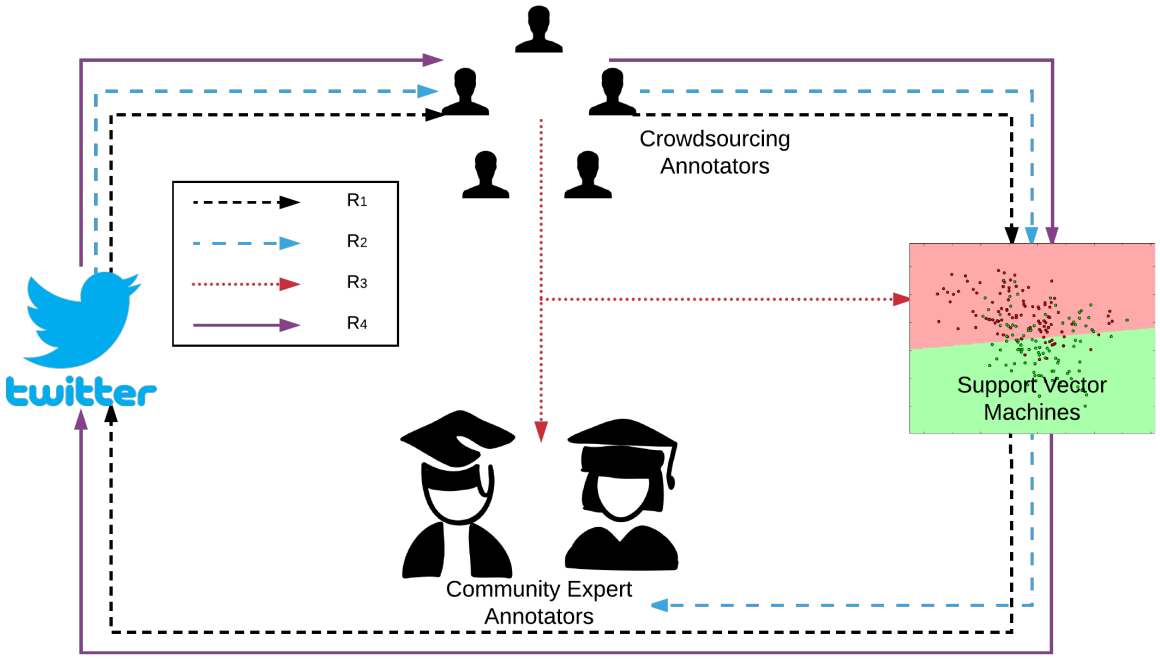

Figure 3.10: Summary of experiments in Liu et al. [7].

Our iterative process used two types of classifiers: rule-based classifiers (C_0 and C_4) and support vector machines (C_1, C_2, C_3 and C_5). In between, there are four rounds of human annotations that progressively grow our labeled dataset in order from R1 to R4 as illustrated in Figure 3.10. Among them, R1, R2, and R4 are crowdsourced annotations, and R3 is contributed by experts in the local

community. We used **unanimous agreement** to determine the ground truth label based on five annotations per tweet, which was then used for training a series of SVM classifiers. This strategy follows the same approach as in our earlier work [6].

Figure 3.11 introduces the workflow of our experiments. Specifically, our classification framework started with a simple rule-based classifier (C_0), followed by three rounds of annotations (two rounds of crowdsourcing annotations R1 and R2, and one round of community annotations R3) and three SVM classifiers (C_1, C_2 and C_3) iteratively to continuously increase the precision measures. We further improved the recall by conducting an additional round of human annotations ($C_4 \rightarrow$ R4) and achieved a more robust SVM classifier (C_5) at the end of the pipeline in order to detect job-related tweets from social media.

3.3.3 Experiment Details to Extract Job-Related Tweets

Annotation Summary

Table 3.10 summarizes the results of our crowdsourced annotation rounds (R1, R2, and R4). For the tweets which had not been unanimously labeled in R1 and R2, local community members (of Rochester, NY) were instructed to annotate them in R3.

Round	**Number of agreements among 5 crowdsourcing annotators**					
	job-related			**not job-related**		
	3	**4**	**5**	**3**	**4**	**5**
R1	104	389	**1,027**	82	116	**270**
R2	140	287	**721**	68	216	**2,568**
R4	214	192	**338**	317	414	**524**

Table 3.10: Summary of crowdsourced annotations (R1, R2 and R4).

To assess the labeling quality of multiple annotators in the crowdsourced annotation rounds (R1, R2 and R4), we calculated Fleiss' kappa [224] and Krippendorff's alpha [225] using an online calculator *Inter-Rater Agreement with multiple raters and variables* [226] to assess inter-annotator reliability among the five annotators of each HIT. And then we calculated the average and standard deviation of inter-annotator scores for multiple HITs per round. Table 3.11 records the inter-annotator agreement scores in three rounds of crowdsourced annotations.

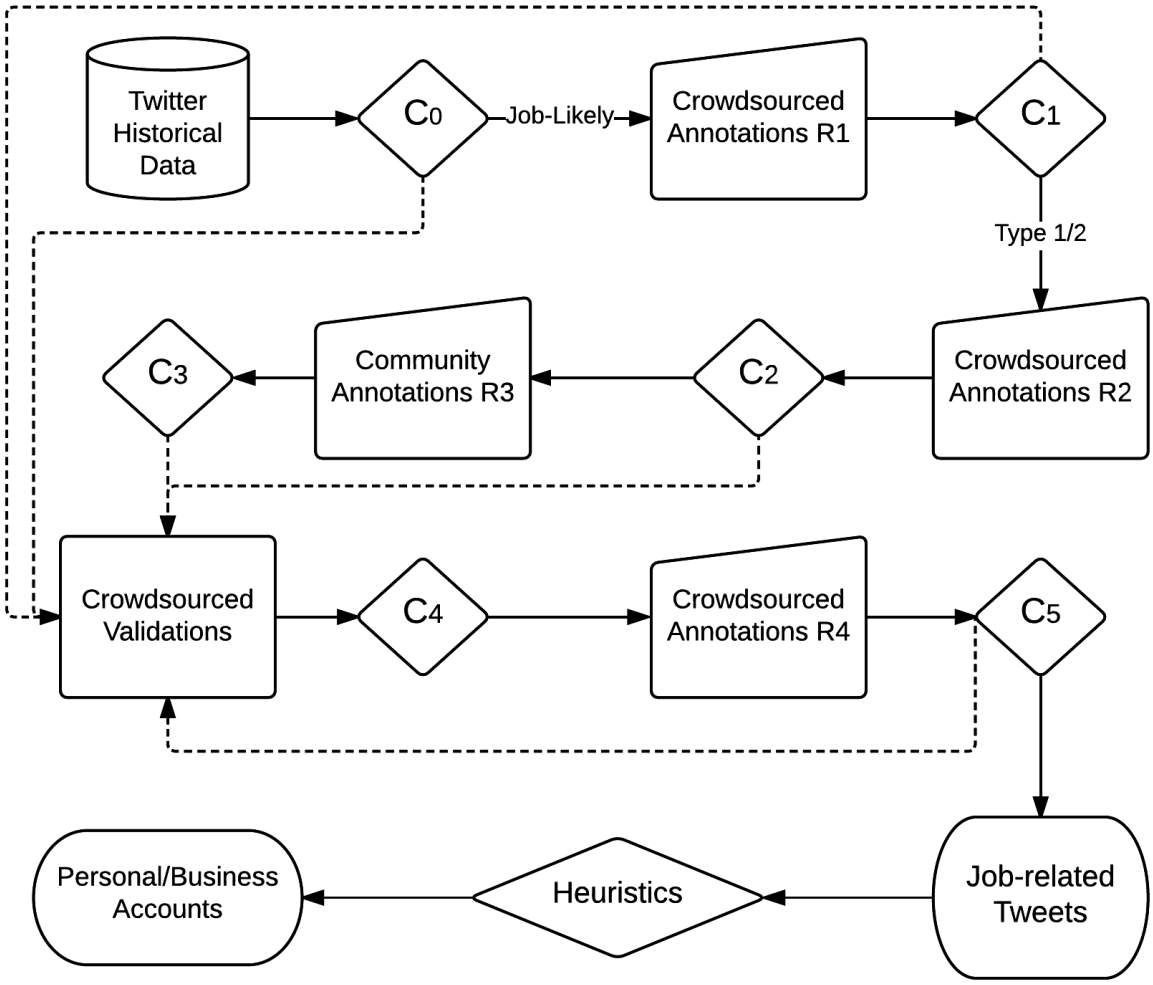

Figure 3.11: Our humans-in-the-loop framework collects labeled data by alternating between human annotation and automatic prediction over multiple rounds. Each diamond represents an automatic classifier (**C**), and each trapezoid represents human annotations (**R**). Each classifier filters and provides machine-predicted labels to tweets that are published to human annotators in the following annotation round. The human-labeled tweets are then used as training data for the next learning round. We use two types of classifiers: rule-based classifiers (C_0 and C_4) and support vector machines (C_1, C_2, C_3 and C_5). This framework serves to reduce the amount of human effort needed to acquire large amounts of high-quality labeled data.

The inter-annotator agreement between the two expert annotators from the local community was assessed using Cohen's kappa [220] as $\kappa = 0.803$ which indicates strong agreement. Their joint

Round	Fleiss' kappa	Krippendorf's alpha
R1	0.62 ± 0.14	0.62 ± 0.14
R2	0.81 ± 0.09	0.81 ± 0.08
R4	0.42 ± 0.27	0.42 ± 0.27

Table 3.11: Inter-annotator agreement performance for our three rounds of crowdsourced annotations (R1, R2 and R4). Average ± stdev agreements are *Good*, *Very Good* and *Moderate* [11], respectively.

efforts corrected more than 90% of the tweets on which crowdsourcing workers in R1 and R2 disagreed.

We observe in Table 3.11 that annotators in R2 achieved the highest average inter-annotator agreement and the lowest standard deviation compared to the other two rounds, suggesting that tweets in R2 have the highest level of confidence of being related to work/employment. As shown in Figure 3.11, the annotated tweets in R1 are the outputs from C_0, the tweets in R2 are from C_1, and the tweets in R4 are from C_4. C_1 is a supervised SVM classifier, while both C_0 and C_4 are rule-based classifiers. The higher agreement scores in R2 indicate that a trained SVM classifier can provide more reliable and less noisy predictions than a rule-based model.

Annotators perform differently on tweets extracted by C_1 and C_4. Higher agreement scores in R1 compared to R4 indicate that the rules in C_4 are not as intuitive as those in C_1, and may even introduce ambiguities. For example, the tweets *What a career from Vince young!* (one of C_1 tasks) and *I hope Derrick Rose plays the best game of his career tonight* (one of C_4 tasks) both use *career*, but convey different information: the first tweet was talking about this professional athlete's accomplishments while the second tweet was actually commenting on the game the user was watching. Hence, the tweets that crowdsourcing workers worked on in C_4 are more ambiguous and difficult to annotate than those found in C_1. Considering that, it is not surprising that the inter-annotator agreement scores of R4 are the worst of all rounds.

Table 3.12 shows the tweets that two community annotators corrected in R3. We excluded tweets on which community members disagreed.

Modeling Experiments Step by Step

We trained different SVM classifiers with training data indicated by Table 3.13. We detail our modeling experiments step by step as below.

From R1 + R2	job-related	not job-related
Y Y Y Y N	644	5
Y Y Y N N	185	17
Y Y N N N	57	51
Y N N N N	11	301
Total	**897**	**374**

Table 3.12: Summary of R3 community-based reviewed-and-corrected annotations.

Input Annotated Data	Output SVM Model
R1U	C_1
R1U + R2U	C_2
R1U + R2U + R3U	C_3
R1U + U2U + R3U + R4U	C_5

Table 3.13: Summary of combinations of annotated data to train different SVM classifiers. +: Union operation to combine data items into an united set.

Initial Classifier C_0

In order to identify probable job-related tweets while excluding noisy tweets (such as students discussing homework or school-related activities, or people complimenting others), we defined a simple term-matching classifier with inclusion and exclusion terms as our first step (see Table 3.14).

Before applying filtering rules, we pre-processed each tweet by (1) converting all words to lower cases; (2) stripping out punctuation and special characters; and (3) normalizing the tweets by mapping out-of-vocabulary phrases (such as abbreviations and acronyms) to standard phrases using a dictionary of more than 5,400 slang terms in the Internet[8].

Classifier C_0 consists of two rules: each tweet must contain at least one word in the *Include* lexicon and no words in the *Exclude* lexicon. This filtering yielded over 40,000 matched tweets having at least five words. We call these tweets *job-likely*.

[8] `http://www.noslang.com/`

[9] Describe something awesome in a sense of utter dominance, magical superiority, or being ridiculously good.

Include	job, jobless, manager, boss my/your/his/her/their/at work
Exclude	school, class, homework, student, course finals, good/nice/great job, boss ass[9]

Table 3.14: The lexicons used by C_0 to extract the *job-likely* set.

Crowdsourced Annotation R1

Our conjecture about crowdsourced annotations, based on the experiments and conclusions from [33], is that non-expert contributors could produce annotations of quality comparable to gold standard annotations from experts. And it is similarly effective to use the labeled tweets with high inter-annotator agreement among multiple non-expert annotators from crowdsourcing platforms to build robust machine models as doing so on expert-labeled data.

We randomly chose 2,000 *job-likely* tweets and split them equally into 50 subsets of 40 tweets each. In each subset, we additionally randomly duplicated five tweets in order to measure intra-annotator agreement. We then constructed Amazon Mechanical Turk (AMT)[10] Human Intelligence Tasks (HITs) to collect reference annotations from crowdsourcing workers. We assigned 5 crowdworkers to each HIT—this is an empirical scale for crowdsourced linguistic annotation tasks suggested by previous studies [34, 227]. Crowdsourcing workers were required to live in the United States and have approval ratings of 90% or better. They were instructed to read each tweet and answer the following question "*Is this tweet about job or employment?*" Workers were allowed to work on as many distinct HITs as they liked.

We paid each worker $1.00 per HIT and gave extra bonuses to those who completed multiple HITs. We rejected workers who did not provide consistent answers to the duplicate tweets in each HIT. Before publishing the HITs to crowdsourcing workers, we consulted with Turker Nation[11] to ensure that we treat and compensate workers fairly for their requested tasks.

Given the sensitive nature of this work, we anonymized all tweets to minimize any inadvertent disclosure of personal information (@names) or cues about an individuals online identity (URLs) before publishing tweets to crowdsourcing workers. We replaced @names with $@SOMEONE$, and recognizable URLs with $HTTP://LINK$. No attempt was ever made to contact or interact with any user.

[10] `https://www.mturk.com/mturk/welcome`

[11] `http://www.turkernation.com`

This labeling round yielded 1,297 tweets labeled with unanimous agreement among five workers, i.e. five workers gave the same label to one tweet—1,027 of these were labeled *job-related*, and the rest 270 were *not job-related*. They composed the first part of our human-annotated dataset, which we named **Part-1**.

Training Classifier C_1

Feature Preparation We relied on a feature space of n-grams (unigrams, bigrams and trigrams) for training. Due to the noisy nature of Twitter, where users frequently write short, informal spellings and grammars, we pre-processed input data as follows. We: (1) utilized a *Twokenizer* specially trained on Twitter texts [222] to tokenize raw messages, (2) completed stemming and lemmatization using WordNet Lemmatizer [223].

Parameter Selection Considering the class imbalance present in the training dataset, we selected optimal learning parameters by grid-searching on a range of class weights for the positive (job-related) and negative (not job-related) classes, and then chose the estimator that optimized F1 score, using 10-fold cross validation.

Classifier C_1 In the *Part-1* set, there are 1,027 job-related and 270 not-job-related tweets. To construct a balanced training set for C_1, we randomly chose 757 tweets outside the *job-likely* set (i.e., which were classified as negative by C_0). Admittedly, these additional samples do not necessarily represent the true negative tweets (*not job-related*), as they have not been manually checked.

Crowdsourced Annotation R2

We conducted the second round of labeling on a subset of C_1-predicted data to evaluate the effectiveness of the aforementioned classifier C_1 and collect more human labeled data to help build a class-balanced labeled set (for training more robust models).

After separating positive- and negative-labeled (*job-related* vs. *not-job-related*) tweets, we sorted each class in descending order of their confidence scores. We then spot-checked the tweets to estimate the frequency of job-related tweets as the confidence score changes. We discovered that among the top-ranked tweets in the positive class about half, and near the separating hyperplane (i.e., where the confidence scores are near zero) almost none, are truly job-related.

We randomly selected 2,400 tweets from those in the top 80th percentile of confidence scores in positive class (*Type-1*)[12]. The *Type-1* tweets are automatically classified as positive, but some of them may in truth not be job-related. Such tweets are the ones on which C_1 fails, though C_1 is very confident about them. We also randomly selected 800 tweets from those having confidence scores that are positive but close to zero, and another 800 tweets from the negative side (*Type-2*). These 1,600 tweets have very low confidence scores, representing those that C_1 cannot clearly distinguish. Thus, the C_1 prediction results of the *Type-2* tweets have a high chance being erroneous. Hence, we considered both the clearer core and at the gray zone periphery of the confidence space.

Crowdworkers were again asked to annotate this combination of *Type-1* and *Type-2* tweets in the same fashion as in R1. Table 3.15 records annotation details.

R2	**Number of agreements among 5 annotators**					
	job-related			**not job-related**		
	3	**4**	**5**	**3**	**4**	**5**
Type-1	129	280	**713**	50	149	**1,079**
Type-2	11	7	**8**	16	67	**1,489**

Table 3.15: Summary of annotations in R2 (showing when 3 / 4 / 5 of 5 annotators agreed).

By grouping the *Type-1* and *Type-2* tweets with unanimous labels in R2 (bold columns in Table 3.15), we had our second part of human-labeled dataset (**Part-2**).

Training Classifier C_2

Combining *Part-1* (from R1) and *Part-2* (from R2) data into one training set—4,586 annotated tweets with perfect inter-annotator agreement (1748 job-related and 2838 not-job-related tweets), we trained the classifier C_2.

Community Annotation R3

Having conducted two rounds of crowdsourced annotations, we noticed that crowdworkers could not reach consensus on a number of tweets. This observation intuitively suggests that non-expert

[12]Note: it is different from the common terms Type I and type II errors in statistical hypothesis testing practice.

annotators may have diverse interpretations about the topic. Table 3.16 provides examples (selected from both R1 and R2) of tweets over six possible inter-annotator agreement combinations.

Crowdsourced Annotations Y/N	**Sample Tweet**
Y Y Y Y Y	Really bored....., no entertainment at work today
Y Y Y Y N	two more days of work then I finally get a day off.
Y Y Y N N	Leaving work at 430 and driving in this snow is going to be the death of me
Y Y N N N	Being a mommy is the hardest but most rewarding job a women can have #babyBliss #babybliss
Y N N N N	These refs need to DO THEIR FUCKING JOBS
N N N N N	One of the best Friday nights I've had in a while

Table 3.16: Inter-annotator agreement combinations and sample tweets. Y represents *job-related* and N represents *job-related*.

Two experts from the local community with prior experience in employment were introduced to review tweets on which crowdworkers disagreed and provided their labels. The tweets with unanimous labels in two rounds of crowdsourced annotations were not re-annotated by experts because their unanimous votes are hypothesized to be reliable as experts' labels.

Thus, we have our third part of human-annotated data (**Part-3**): tweets reviewed and corrected by the community annotators.

Training Classifier C_3

Combining *Part-3* with all unanimously labeled data from the previous rounds (*Part-1* and *Part-2*) yielded 2,645 gold-standard-labeled job-related and 3,212 not job-related tweets. We trained C_3 on

this entire training set.

Crowdsourced Validation of C_0, C_1, C_2 and C_3

To evaluate uniformly the models in different stages—including the initial rule-based classifier C_0—we adopted a post-hoc evaluation procedure: We sampled 400 distinct tweets that have not been used before from the data pool labeled by C_0, C_1, C_2 and C_3 respectively (there is no intersection between any two sets of samples). We used these four classifiers to label this combined test set of 1600 tweets. We then asked crowdsourced workers to validate this set, in a manner identical to the previous rounds of crowdsourced annotations (R1 and R2). We took the majority label (where at least 3 out of 5 crowdsourced workers agreed) as the reference label for each tweet.

Table 3.17 displays the classification results of each model against the reference labels provided by crowdsourcing workers. It shows that C_3 outperforms C_0, C_1, and C_2.

Model	Class	P	R	F1
C_0	job	0.72	0.33	0.45
	notjob	0.68	0.92	0.78
	avg / total	*0.70*	*0.69*	*0.65*
C_1	job	0.79	0.82	0.80
	notjob	0.88	0.86	0.87
	avg / total	*0.85*	*0.84*	*0.84*
C_2	job	0.82	0.95	0.88
	notjob	0.97	0.86	0.91
	avg / total	*0.91*	*0.90*	*0.90*
C_3	job	0.83	0.96	0.89
	notjob	0.97	0.87	0.92
	avg / total	*0.92*	*0.91*	*0.91*

Table 3.17: Crowdsourced validations of samples identified by models C_0, C_1, C_2 and C_3.

Crowdsourced Annotation R4

Even though C_3 achieves the highest performance among the four classifiers we test, it has room for improvement. We manually checked the tweets in the test set that were incorrectly classified as *not-job-related* and focused on the language features we ignored in preparation for the model

training. After performing some pre-processing on the tweets in the false negative and true positive groups from the above tests, we ranked and compared their distributions of word frequencies. These two rankings reveal the differences between the two categories (false negative vs. true positive) and help us discover some signal words that were prominent in false negative group but not in true positive—if our trained models are able to recognize these features when forming the separating boundaries, the false negative rates would decrease and overall performance would further improve.

Our fourth classifier C_4 is rule-based again, in order to select more potentially job-related tweets, especially those would have been misclassified by our trained models. The lexicons in C_4 include the following signal words: *career*, *hustle*, *wrk*, *employed*, *training*, *payday*, *company*, *coworker* and *agent*.

We ran C_4 on our data pool and randomly selected 2,000 tweets that were labeled as positive by C_4 and never used previously (i.e., not annotated, trained or tested in C_0, C_1, C_2, or C_3). We published these tweets to crowdsouring workers using the same settings of R1 and R2. The tweets with unanimously agreed labels in R4 form the last part of our human-labeled dataset (**Part-4**).

Training Classifier C_5

Aggregating separate parts of human-labeled data (*Part-1* to *Part-4*), we obtained an integrated training set with 2,983 job-related tweets and 3,736 not-job-related tweets and trained C_5 upon it. We tested C_5 using the same data from the crowdsourced validation phase (1,600 tested tweets) and discovered that C_5 beat the performances of the other models (Table 3.18).

Model	Class	P	R	F1
C_5	job	0.83	0.97	0.89
	notjob	0.98	0.87	0.92
	avg / total	*0.92*	*0.91*	*0.91*

Table 3.18: Performances of C_5.

Table 3.19 lists the top 15 features for both classes in C_5 with their corresponding weights. Positive features (job-related) unearth expressions about personal job satisfaction (*lovemyjob*) and announcements of working schedules (*day off*, *break*) beyond our rules defined in C_0 and C_4. Negative features (not job-related) identify phrases to comment on others' work (*your work*, *amazing job*, *awesome job*, *nut job*) though they contain "work" or "job," and show that school- or game-themed messages (*college career*, *play*) are not classified into the job class, which meets our original

intention.

job-related	weights	not job-related	weights
job	1.77	your work	-0.61
manager	1.71	like it	-0.60
work	1.69	amazing job	-0.59
wrk	1.44	did	-0.55
payday	1.23	nut	-0.45
my bos	1.06	nut job	-0.45
jobs	0.83	bos as	-0.43
lovemyjob	0.81	play	-0.41
at work	0.81	awesome job	-0.38
working	0.75	college career	-0.37
my career	0.74	high	-0.36
day off	0.73	doing	-0.35
boss	0.73	hustle	-0.35
service	0.71	you guy	-0.33
break	0.70	love your	-0.33

Table 3.19: Top 15 features for both classes of C_5.

Table 3.20 combines the performance results in Table 3.17 and 3.18, covering different models (C_0, C_1, C_2, C_3 and C_5)[13] on a 1600-tweets ad-hoc test set.

End-to-End Evaluation

The class distribution in the machine-labeled test data is roughly balanced, which is not the case in real-world scenarios, where not-job-related tweets are much more common than job-related ones.

We proposed an end-to-end evaluation: to what degree can our trained automatic classifiers (C_1, C_2, C_3 and C_5) identify job-related tweets in the real world? We introduced the *estimated effective recall* under the assumption that for each model, the error rates in our test samples (1,600 tweets) are proportional to the actual error rates found in the entire one-year data set which resembles the real world. We labeled the entire data set using each classifier and defined the estimated effective

[13] C_4 is not in the table because we did not test it in our experiments.

Model	Class	P	R	F1
$\mathbf{C_0}$	job	0.72	0.33	0.45
	notjob	0.68	0.92	0.78
	avg / total	*0.70*	*0.69*	*0.65*
$\mathbf{C_1}$	job	0.79	0.82	0.80
	notjob	0.88	0.86	0.87
	avg / total	*0.85*	*0.84*	*0.84*
$\mathbf{C_2}$	job	0.82	0.95	0.88
	notjob	0.97	0.86	0.91
	avg / total	*0.91*	*0.90*	*0.90*
$\mathbf{C_3}$	job	0.83	0.96	0.89
	notjob	0.97	0.87	0.92
	avg / total	*0.92*	*0.91*	*0.91*
$\mathbf{C_5}$	job	0.83	0.97	0.89
	notjob	0.98	0.87	0.92
	avg / total	*0.92*	*0.91*	*0.91*

Table 3.20: Crowdsourced validations of samples identified by models C_0, C_1, C_2, C_3 and C_5, with the best model highlighted in red.

recall $\hat{R}$ for each classifier as

$$\hat{R} = \frac{Y \cdot N_t \cdot R}{Y \cdot N_t \cdot R + N \cdot Y_t \cdot (1 - R)}$$

where Y is the total number of the classifier-labeled job-related tweets in the entire one-year data set, N is the total of not-job-related tweets in the entire one-year data set, Y_t is the number of classifier-labeled job-related tweets in our 1,600-sample test set, $N_t = 1,600 - Y_t$, and R is the recall of the job class in our test set, as reported in Tables 3.17 and 3.18.

Table 3.21 shows that C_5 had the best effective recall score, though here there is still room for improvement.

3.3.4 Determining Sources of Job-Related Tweets

Through observation we noticed some patterns like:

"*Panera Bread: Baker - Night (#Rochester, NY) HTTP://URL #Hospitality #Veter-*

Models	C_1	C_2	C_3	C_5
Y	115,696	195,442	190,471	233,187
N	6,990,633	6,910,887	6,915,858	6,873,142
$\mathbf{Y_t}$	512	691	707	729
$\mathbf{N_t}$	1,088	909	893	871
R	0.82	0.95	0.96	0.97
$\hat{\mathbf{R}}$	0.14	0.41	0.46	0.57

Table 3.21: Estimated effective recalls for different trained models (C_1, C_2, C_3 and C_5) to identify job-related tweets in real world settings.

anJob #Job #Jobs #TweetMyJobs"

in the job-related tweets. Nearly every job-related tweet that contained at least one of the following hashtags: *#veteranjob*, *#job*, *#jobs*, *#tweetmyjobs*, *#hiring*, *#retail*, *#realestate*, *#hr* also had a URL embedded. We counted the tweets containing only the listed hashtags, and the tweets having both the queried hashtags and embedded URL, and summarized the statistics in Table 3.22. By spot checking we found such tweets always led to recruitment websites. This observation suggests that these tweets with similar "hashtags + URL" patterns originated from business agencies or companies instead of personal accounts, because individuals are unlikely to post recruitment advertising.

	hashtag only	**hashtag + URL**	**%**
#veteranjob	18,066	18,066	100.00
#job	79,359	79,326	99.96
#jobs	59,882	59,864	99.97
#tweetmyjobs	39,007	39,007	100.00
#hiring	622	619	99.52
#retail	17,107	17,105	99.99
#realestate	113	112	99.12
#hr	406	405	99.75

Table 3.22: Counts of tweets containing the queried hashtags only, and their subsets of tweets with URL embedded.

This motivated a simple heuristic that appeared surprisingly effective at determining which kind of accounts each job-related tweet was posted from: if an account had more job-related tweets matching the "hashtags + URL" patterns than tweets in other topics, we labeled it a *business*

account; otherwise it is a *personal* account. We validated its effectiveness using the job-related tweets sampled by the models in crowdsourced evaluations phase. It is essential to note that when crowdsourcing annotators made judgment about the type of accounts as *personal* or *business*, they were shown only one target tweet—without any contexts or posts history which our heuristics rely on.

Table 3.23 records the performance metrics and confirms that our heuristics to determine the sources of job-related tweets (*personal* vs. *business* accounts) are consistently accurate and effective.

From	Class	P	R	F1
$\mathbf{C_1}$	personal	1.00	0.98	0.99
	business	0.98	1.00	0.99
	avg/total	*0.99*	*0.99*	*0.99*
$\mathbf{C_2}$	personal	1.00	0.99	0.99
	business	0.99	1.00	0.99
	avg/total	*0.99*	*0.99*	*0.99*
$\mathbf{C_3}$	personal	1.00	0.99	0.99
	business	0.99	1.00	0.99
	avg/total	*0.99*	*0.99*	*0.99*
$\mathbf{C_5}$	personal	1.00	0.99	0.99
	business	0.99	1.00	0.99
	avg/total	*0.99*	*0.99*	*0.99*

Table 3.23: Evaluation of heuristics to determine the type of accounts (personal vs. business), job-related tweets sampled by different models in Table 3.17.

Count of Labels		Human	Machine
Topic	job	2,978	233,187
	notjob	3,736	6,873,142
	NA	842	–
Source	personal	1,357	7,025,203
	business	232	81,126
	NA	5,966	–

Table 3.24: Statistics of our Twitter Job/Employment Corpus.

3.3.5 Twitter Job/Employment Corpus

Our humans-in-the-loop active learning framework produced a series of classification models. Among all models, $\mathbf{C_5}$ performed the best (highlighted in red in Table 3.20) in detecting (non) job-related tweets. Further, we applied heuristics to separate accounts posting job-related tweets into personal and business groups automatically based on linguistic features of each Twitter account's tweets.

Table 3.24 shows the main statistics of our Twitter Job/Employment Corpus w.r.t the topic and source labels provided by human and machine (C_5) respectively.

Additionally, though our classification models are support vector machines built for work and employment domain, this framework can integrate human inputs with different machine learning algorithms and models to solve other similar open-domain problems that lack high-quality labeled ground truth data.

Chapter 4

Population Label Distribution Learning

As machine learning (ML) plays an ever increasing role in commerce, government, and daily life, reports of bias in ML systems against groups traditionally underrepresented in computing technologies have also increased. The problem appears to be extensive, yet it remains challenging even to fully assess the scope, let alone fix it. A fundamental reason is that ML systems are typically trained to predict one correct answer or set of answers; disagreements between the annotators who provide the training labels are resolved by either discarding minority opinions (which may correspond to demographic minorities or not) or presenting all opinions flatly, with no attempt to quantify how different answers might be distributed in society. Label distribution learning associates for each data item a probability distribution over the labels for that item. While such distributions may be representative of minority beliefs or not, they at least preserve diversities of opinion that conventional learning hides or ignores and represent a fundamental first step toward ML systems that can model diversity. We introduce a strategy for learning label distributions with only five-to-ten labels per item—a range that is typical of supervised learning datasets—by aggregating human-annotated labels over multiple, similarly rated data items. Our results suggest that specific label aggregation methods can help provide reliable, representative predictions at the population level.

4.1 Problem Statement

The *population label distribution learning problem* is to learn to predict the distribution of labels $\mathbf{y}$ among a population of annotators for each test set data item $\mathbf{x}$, given a collection of training data items $(\mathbf{x}_i)_{i\in\{1,\ldots,n\}}$ and a corresponding collection of label distribution *raw estimates* $(\hat{\mathbf{y}}_i)_{i\in\{1,\ldots,n\}}$, based on the normalized *empirical label distributions*, i.e., the distributions of the annotations received for each data item. Note that, here, we assume these distributions are multinomial samples of the underlying population of annotator's *true label distribution* $(\mathbf{y}_i)_{i\in\{1,\ldots,n\}}$, and that the each raw estimate was obtained by randomly choosing an annotator and then asking that annotator to choose a label, then repeating this process m times, where m is a parameter of the sampling process.

One example of a label set that supports this problem definition came from an effort to model Twitter discourse on life trajectories. When inspecting annotators' answers to a question that identifies employment transition events, we observed that when there was disagreement, it was often for good reason.

Figure 4.1 shows the label distributions over the the jobQ3MT+ label set (see more details below in Section 4.3). These histograms of labels (one histogram per data item) appear to cluster into approximately eight categories, where the tweets in each seemed to be similarly rated. Group 1 (red) distributions have most of their mass on *Getting hired/job seeking* and *None of the above, but job-related*, with tweets talking about plans to get a job (e.g., *really want a job*, *dont put that on ur resume for a minimum wage job*) or the process of getting a job. Group 2 (cyan) has almost all the mass exclusively on *Getting hired/job seeking* (e.g., *got the job*). Group 3 (brown) clusters around *Complaining about work* and *Going to work*, suggesting a topic about complaining about having to go to work. Group 4 (green) are a set of tweets complaining about work while at work. Groups 5 and 6 (blue and orange) have their peaks on *None of the above, but job-related* and *Not job-related*. Group 6 (where *Not job-related* was more frequent than *None of the above*) were mostly about road work. Group 7 seemed to contain cases where work was mentioned, but not central (e.g., *Today at work I learned about...*) or used "work" or "job" metaphorically, though there exist some clear *None of the above, but job-related* tweets, like *Perks of working overnight: donuts fresh out of the fryer*.

As to why such clustering happens, Zhang et al. [109], on a different dataset, noticed similar clustering patterns. We note that any k-choice annotation task effectively reduces the full breadth of interpretations encoded in each data item x to one of only k choices; We theorize that the act

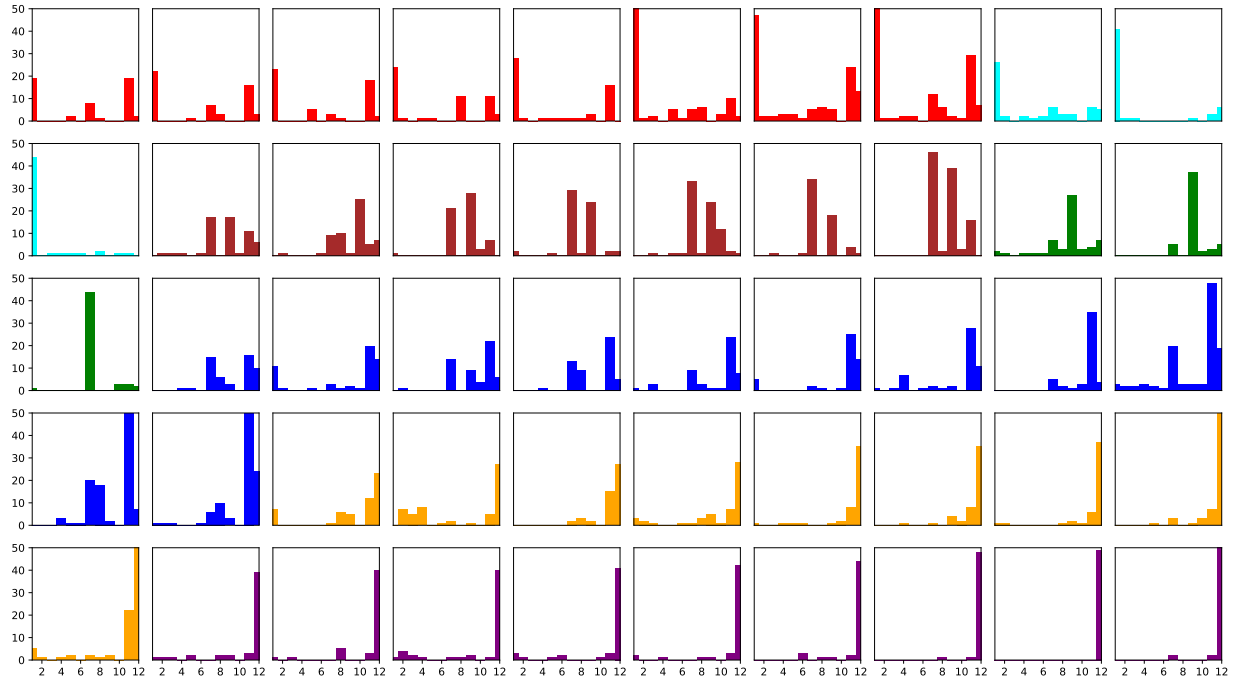

Figure 4.1: Each histogram above represents the label distribution of a lone data item in the jobQ3MT+ data set. The X-axis ranges from 1 to 12, matching the Q3 choices in Example 4.3.1. The Y-axis denotes the label counts. Similar distributions are grouped by color: 1-8 red, 9-11 cyan, 12-18 brown, 19-21 green, 22-32 blue, 33-41 orange, and 42-50 purple.

.

of annotation reduces not only the interpretive domain of the each data item, but also the social, experiential, and cognitive factors, such as disparities in experience and knowledge, that drive annotator disagreement. Thus, the number p of *distinct* ground truth label distributions resulting from any annotation task are also limited, and the set of all annotations for any given data item is (assuming annotators are selected i.i.d. from the population of annotators) a sample from one of the p distinct ground truth distributions. For the sake of brevity we subsequently refer to this tentative explanation as the **clustering theory**.

4.1.1 Label Probability Distribution

Instead of inputting a single label associated with a data item, we introduce a probabilistic method which inputs to machine learning models a label distribution—class labels with corresponding probabilities—collected from multiple annotators. This approach has the advantage that subjec-

tivity for multiple choices among multiple annotators is taken into consideration and represented for learning purpose. The intuition behind this idea is straightforward: we assume that crowdsourcing workers who get access to and take participation into the annotation tasks are independent and identically distributed in the global crowdsourcing community. These unspecified annotators can represent the natural distribution of overall population and their understandings toward subjective topics and themes.

4.2 LDL Algorithms

Our approach to label distribution learning on populations consists of two stages. First, we use unsupervised learning to convert the raw label distribution estimates $(\hat{\mathbf{y}}_i)_{i\in\{1,\ldots,n\}}$, into *refined estimates* $(\hat{\mathbf{y}}'_\mathbf{i})_{i\in\{1,\ldots,n\}}$, by aggregating over similarly related data items. Next, we perform supervised learning on the refined label distributions with unstructured text features and conduct comparative experiments. We discuss each stage below.

4.2.1 Clustering Algorithms for Estimating Ground Truth

The unsupervised learning algorithms we consider here are consistent, to varying degrees, with the clustering theory.

The (finite) multinomial mixture model **F** most directly simulates the sampling process of collecting annotations from crowdsourcing community according to the cluster theory. It assumes that the empirical label distributions are generated by: (1) drawing a multinomial distribution π according to a Dirichlet prior over p elements which correspond to the hypothesized number of true label distributions, denoted as $\pi \sim \mathrm{Dir}(p, \gamma = 75)$, where γ is the prior's (symmetric) hyperparameter (and higher numbers tend to produce lower entropy multinomials); (2) drawing multinomial distributions $\phi_1, \ldots \phi_p \sim \mathrm{Dir}(d, \gamma = 0.1)$, where d is the number of label choices; (3) for each data item, we (3a) choose $i \sim \pi$ and (3b) m label classes according to ϕ_i. Thus, according to the clustering theory, the most likely cluster distribution ϕ_j for each data item should be a good estimate of the ground truth label distribution: $\phi_j \approx \mathbf{y}_i$. We learn the model **F** via a variational Bayes algorithm [228], adapted from `https://github.com/bnpy/bnpy`.

Next come two variants of **F**. The Dirichlet process multinomial mixture model (abbreviated as **DP**) is a non-parametric version of model **F**. Instead of choosing p multinomial models from a Dirichlet

prior before generating the data, **DP** starts with two multinomial models $\phi_1, \phi_2 \sim \text{Dir}(d, 0.1)$. Then, for each new data item, it draws from the current set of multinomial models in approximate proportion to the number of times each has been previously drawn OR draws a new multinomial model (with weight proportional to $\gamma = 50$). We use a variational Bayes algorithm to learn this model too. The main purpose for including **DP** here is to test whether in this setting nonparametric methods outperform parametric ones using standard model-selection criteria.

M is a multinomial mixture model without any Dirichlet priors. This rather simple model can be learned using EM [229], however it lacks the regularization and adaptability that the Dirichlet priors can provide. We expect this model to underperform the others as a result.

In contrast to the previous models, we chose the Gaussian mixture model **G** as a weak alternate hypothesis of sorts. Rather than simulate the sampling process of crowdsourced annotations as the multinomial distributions do, these distributions capture the variance in a population of samples. Additionally, it captures covariance between the labels; these should be close to zero in single label settings (or settings where the vast majority of annotators provide only one label per item). We use EM to learn this model[1] .

Finally, **L** is latent Dirichlet allocation (LDA) [230], adapted from `https://radimrehurek.com/gensim/`. LDA is a generative topic model which is widely used in natural language processing area [230]. A collection of documents can be described as a mixture of various latent topics with a sparse Dirichlet prior distribution, where each topic, associated with a small set of descriptive words, could be assigned to the document via LDA probabilistically[2]. Though LDA is not designated for clustering tasks, we can obtain cluster-like latent classes in the modeling process. In contrast to **F**, which chooses a single class distribution π for all data items, **L** chooses a new distribution π_i for each item i and then for each label chooses a new distribution in $\{\phi_1, \ldots, \phi_p\}$ according to π_i. Thus, for each data item, each instance of the labels from LDA represents a true mixture of all the generating distributions. We can thus "assign" to i the most likely ϕ_j according to π_i.

[1] `http://scikit-learn.org/stable/modules/mixture.html`

[2] We treat each document as a string concatenation of labels collected from multiple annotators in our study, for example, one data point would be "AABCCD" when aggregating six human labels into a concatenated string. We denote this strategy as $\boldsymbol{LDA}_{label}$.

4.2.2 Supervised Learning for Predicting Label Distributions

We train supervised-learning-based classifiers using refined label distributions obtained from the various unsupervised learning algorithms described in Section 4.2.1. As a pre-processing step, we retain the most common 20,000 words in the training set, pad the sentence with up to 1,000 tokens, and then embed each word into a 100-dimension real-valued vector using the GloVe pre-trained word embeddings trained on a Twitter corpus with 2B tweets [231].

We consider two neural network models. One is built on a 1D convolutional neural network (denoted as **CNN**) which was designed for sentence or tweet classification [193]. There are three max pool/convolution layers, followed by a dropout layer and a softmax layer. We use the *Adam* optimizer to minimize the loss function [232], with a batch size as 32 and 25 epochs. Another model type used to address prediction problems is the encoder-decoder sequence-to-sequence model which uses recurrent neural networks (**LSTM**). The encoder outputs a fixed-length encoding of the input text, and the decoder, followed by a *Dense* output layer, predicts the output sequence. We applied the same optimizer, batch size and epoch number as in the CNN models.

In the process of training both types of models, we use *softmax* function: $\frac{exp(z_i)}{\Sigma_t exp(z_t)}$, to transform the output of the penultimate layer $\mathbf{z}$ into a probability distribution. We use *Kullback-Leibler (KL) divergence*, a standard measure of the difference between the "true" (in our case the refined estimate) probability distribution $\hat{\mathbf{y}}'$ and a predicted estimator $\tilde{y}$: $\mathrm{D}_{KL}(\hat{y}', \tilde{y}) = \sum_i P(\hat{y}' = i) \frac{\log P(\hat{y}'=i)}{\log P(\tilde{y}=i)}$, as the loss function for backpropagation, as it is a principled choice for approximating the probability distributions [233].

4.3 Data and Labels

There are several publicly available human-labeled datasets, most of which only contain the crowd-sourced labels without complete textual contents and contexts (for building language based models), thus are not suitable to our population label distribution learning task.

We consider two corpora in our experiments, each consisting of 2,000 Twitter posts (tweets) with crowdsourced labels, one related to work (mentioned in Section 4.1), the other to suicide. Before this, we performed preliminary research on a broader variety of human-labeled data (including [234, 235, 236, 237]) and decided that these two corpora adequately represent the others and were similar in media source (both are collected from Twitter), but different in the domain of discourse.

Thus, they provide an informative basis for comparison. Our institutional review board determined that our work did not fall under federal or institutional guidelines as human subjects research. To privatize the data, we replaced all mentions of usernames with "@SOMEONE" and URLs with "http://URL," and adhered to Twitter's developer policy [238]. Table 4.1 summarizes basic properties of the labels we collected for these two corpora.

ID	Label Set	#Items	#Choices /item	#Workers	#Labels	Density	MVTD	RMSD
1	jobQ1FE	2,000	5	171	10,000	5.00	0.37	0.21
2	jobQ1MT	2,000	5	1,014	12,202	6.10	0.17	0.10
3	jobQ1BOTH	2,000	5	1,185	22,202	11.10	0.29	0.16
4	jobQ1MT+	50	5	249	2,969	59.38	0.43	0.22
5	jobQ2FE	2,000	5	171	10,000	5.00	0.28	0.16
6	jobQ2MT	2,000	5	1,014	12,202	6.10	0.15	0.09
7	jobQ2BOTH	2,000	5	1,185	22,202	11.10	0.23	0.13
8	jobQ2MT+	50	5	249	2,969	59.38	0.34	0.19
9	jobQ3FE	2,000	12	171	10,967	5.48	0.45	0.16
10	jobQ3MT	2,000	12	1,014	12,900	6.45	0.28	0.10
11	jobQ3BOTH	2,000	12	1,185	23,867	11.93	0.40	0.14
12	jobQ3MT+	50	12	249	3,196	63.92	0.41	0.14
13	Suicide	2,000	4	124	13,175	6.59	0.27	0.17

Table 4.1: Basic properties of our crowdsourced label sets. For the job-related data set with three questions *jobQ1/2/3*, *FE* and *MT* represent the labels from the platforms Figure Eight and Amazon Mechanical Turk, respectively. *jobQ1/2/3BOTH* integrates labels from both FE and MT sources into one set. *jobQ1/2/3MT+* denote the additional MT labels used in one experiment setting (*deep split*). **Density** is the average number of labels per data item. **MVTD** (majority-voted-true-class deviation) and **RMSD** (root-mean-square deviation) describe inter-rater reliability across all the tasks and estimate the variety and divergence of human labels in different label sets, motivated by the literature on scale and outlier description [12, 13, 14]. MVTD is the average deviation of the majority-voted label over all data items: MVTD $= 1 - \sum_{i=1}^{n} \max_j\{\hat{y}_{ij}\}/n$. RMSD is the L2 deviation from the average label distribution: RMSD $= \sum_{i=1}^{n} \sqrt{(\hat{\mathbf{y}}_i - \overline{y})^T(\hat{\mathbf{y}}_i - \overline{y})}/n$, where $\overline{y}$ is the average label distribution over all data.

We now discuss annotation tasks in detail to get a better sense of how subjectivity presents itself in labeling tasks and how we operationalize our crowdsourced annotations to prepare for the next modeling phases.

4.3.1 Job-related Annotation

Background

We extend our previous study of job and employment issues in public social media [7], to conduct experiments about subjective employment stage categorization. It is expected that our humans-in-the-loop learning framework largely helps reduce human efforts to annotate data, handles subjectivity among multiple annotators, and provides fine-grained information for downstream applications.

Working-age adults spend more than one-third of their daily time on job-related activities [239]—more than on anything else. Their work conditions and degrees of satisfaction may pose serious health risks and even lead to suicide in the tragic extreme [240, 241]. Extracting job-related discourse is useful for building systems for targeted social networking and recruiting (e.g., recruiters reach out to those who recently lost jobs or intend to move to new positions), job/company/community recommendations (e.g., job search websites recommend job openings or communities based on the user's specific employment status, current or previous employer, or occupation), mental health monitoring system for working people's moods and stress levels (e.g., clinical psychologists understand if people suffer from any mental issues and provide professional intervention and support if necessary), and so on.

This kind of analysis can further monitor job satisfaction/dissatisfaction, and help people strive for positive changes at work and in life [211]. We would like to understand better what people reveal about their work and employment lives in their online messages and how they are interpreted differently by readers and others. For instance, people can get happy if they get job offers and they might feel sad if getting fired. Such career transitions can have huge impacts on mental health for the tweet author him or her self and the broader public.

One challenge in modeling the dynamics of employment is to cover as many distinct situations as possible, but not be so complicated and impractical as to be infeasible for machine learning. Based on manual inspection of a large number of job-related tweets and on models of the relationship between work and wellness found in behavioral studies [242, 243], we drafted a model of the **Job Status Cycle** for job-related discourse from individual accounts (Figure 4.2). Each state in the model has three dimensions: the *point of view*, the *affect*, and the job-related *activity*, in terms of basic level of employment, expressed in the tweet. In general, an individual's job status cycles between employed and jobless/unemployed, with two transitional states in between. There is also a dimension representing under- or over- employment, which is loosely coupled with the employed

status. Note that it is possible to skip the jobless state and go directly to a new job from the current job. Also, it is possible for people to hold multiple jobs at once.

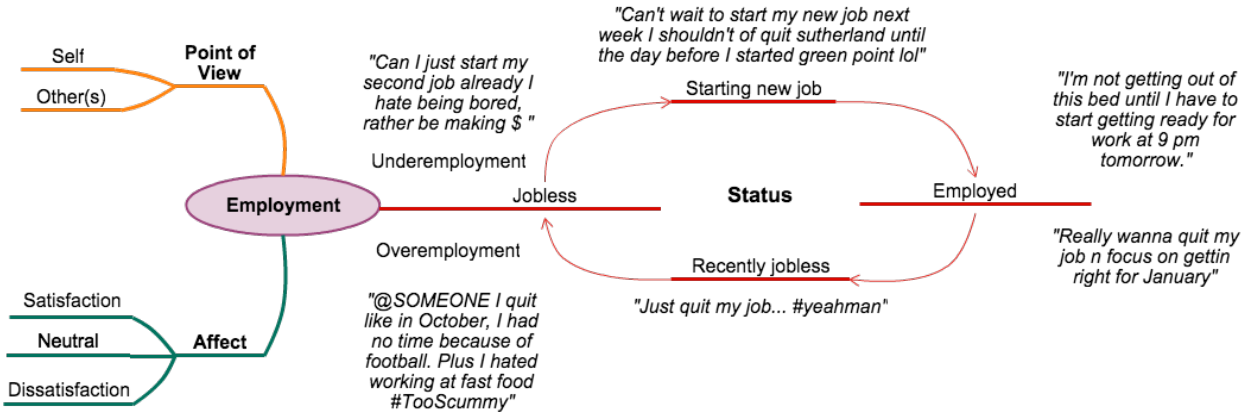

Figure 4.2: The job status cycle, plus example tweets.

Annotation Scheme

We utilize our Twitter Job/Employment Corpus [7] which was previously mentioned in Section 4.1. It contains 2,000 tweets about work and employment that were extracted by the classifier described in the previous section [7].

For each tweet, we asked five crowdworkers from Figure Eight (**FE**, [244]) and Amazon Mechanical Turk (**MT**, [245]) to answer three multiple-choice questions. Example 4.3.1 shows the three questions we asked and their corresponding selections of labels. We denote these label sets *jobQ1/2/3*. To provide some insight into how performance might change with more labels from a more diverse population of labelers and labeling platforms, we first consider FE and MT as two separate label sets, then combine them into a single label set (denoted BOTH).

For each question, our experiments are conducted on two different train/dev/test splits. We first consider a 1000/500/500 split on each of the label sets: Q1, Q2, and Q3 (we call it the **Broad** split). Next, to get a more accurate ground-truth estimate for testing, we randomly selected 50 tweets from our dataset and asked 50 additional MT crowdworkers to label them as instructed. We denote these dense label sets *jobQ1/2/3MT+* and then create 1500/450/50 splits (called the **Deep** split), where the training and development sets are from the BOTH label sets (minus the *jobQ1/Q2/Q3MT+* label set items) and the test sets are from *jobQ1/Q2/Q3MT+*, respectively.

It is important to note that, when we publish the job-related tweets to annotators, we provide some contextual information in addition to the target tweet. We present these contexts in the form of the three tweets preceding and the three succeeding the target tweet from the tweeter's timeline, along with the relative time difference between each context and target tweet. The contextual tweets are not necessarily job-related because people use Twitter to express a wide range of facts and opinions. The contextual information is supposed to reduce the difficulty of the judgment-making process for annotators. We have used such similar design in Homan et al. [5] to annotate distress level and proved its usefulness.

We created multiple-choice questions to categorize the employment stage(s) for the subject in the target tweet and instructed annotators to understand their tasks. We present an example annotation interface below for the tweet *I need a new job. Preferably one that actually pays money.*, with concept definition and annotation instruction.

An Example

> The following definitions describe three employment statuses. Please read them carefully, make your judgments to finish three multiple-choice questions.
>
> **Employed** is defined as:
>
> - Working for pay (salary workers), or profit (self-employed) during the census survey reference week.
> - Working in a family-operate business or farm (at least 15 hours per week without pay).
> - Being temporarily absent from their regular jobs (no matter they were paid or not during the time off) because of vocation, illness, maternity/paternity leave, family/personal obligations, labor dispute, bad weather or other short-term reasons.
>
> **Unemployed** is defined as:
>
> - Not having a job at all during the survey reference week.
> - People made specific efforts to look for a job in the prior 4 weeks, such as: contacting an employer or employment agency, submitting resumes or job applications, placing or answering job advertisements, etc.
> - People were available for work, such as expecting to be recalled from temporary layoff (unless temporarily ill).

Not in labor force is defined as:

- Neither having a job nor looking for one.

Now, using the above concepts you just learned, please read the following target tweet (highlighted in bold and marked with >>>and <<<) to answer multiple-choice questions Q1 through Q3. You can use the three tweets immediately before (marked as -3, -2 and -1) and the three after (+1, +2, +3) the target tweet made by the same user to help make your judgments, where "[day, hour-minute-second]" indicates the time difference between each context tweet and the target tweet you need to label.

MESSAGE ID: 1234567890 DATE: XXXX-YY-ZZ

-3: message 1 [-02, 00-40-35]

-2: message 2 [-01, 14-28-06]

-1: message 3 [-00, 05-17-23]

>>> **I need a new job. Preferably one that actually pays money.** <<<

+1: message 4 [+00, 02-00-59]

+2: message 5 [+00, 17-21-00]

+3: message 6 [+01, 01-05-16]

Q1: Which of the following items best describes the point of view of job/employment-related information in the target tweet?

A. 1st person

B. 2nd person

C. 3rd person

D. Unclear

E. Not job-related

Q2: Which of the following items could best describe the employment status of the subject in the tweet?

A. Employed

B. Unemployed

C. Not in labor force

D. Unclear

E. Not job-related

Q3: Does the subject specifically mention any job/employment transition event in the tweet? (Choose all that apply)

1. Getting hired/job seeking
2. Getting Fired
3. Quitting a job
4. Losing job some other way
5. Getting promoted/raised
6. Getting cut in hours
7. Complaining about work
8. Offering support
9. Going to work
10. Coming home from work
11. None of the above, but job-related
12. Not job-related

Manual Annotations We utilize crowdsourcing to hire multiple annotators (who are usually non-experts) to obtain representative samples of the underlying population distribution, in order to reduce cost and effort, as in Liu et al. [7]. Moreover, crowdsourcing platforms provide us access to annotators with various backgrounds and skill levels, thus facilitating our study about subjectivity issues over a broader population than a study using just one platform would have.

4.3.2 Suicide-related

The *Suicide* tweet set was obtained directly from [6] (introduced earlier in Section 3.2), which contains 2000 tweets about suicide-related discourse. For each data item, five crowdworkers each chose one label that best describes its content, from four possible choices: Ⓐ *Suicidal thoughts*, Ⓑ *Supportive messages or helpful information*, Ⓒ *Reaction to suicide news/movie/music* and Ⓓ *Others*. Experts were invited to the second annotation stage to work on the tweets without unanimous labels provided by five crowdworkers. Thus each tweet can have up to 7 labels, from crowdworkers and experts. We use a 1000/500/500 train/dev/test split (denoted as **Broad** split) in our experiments.

4.4 Experiments

Different labeling strategies lead to different estimated label distributions that are used in the model training stage through the humans-in-the-loop pipeline. We investigate several quantitative methods to aggregate multiple annotations and transmit the aggregated label into downstream supervised classification models.

Figure 4.3 summarizes our experiment framework, which includes data and label collection, unsupervised and supervised modeling phrases, and performance evaluations.

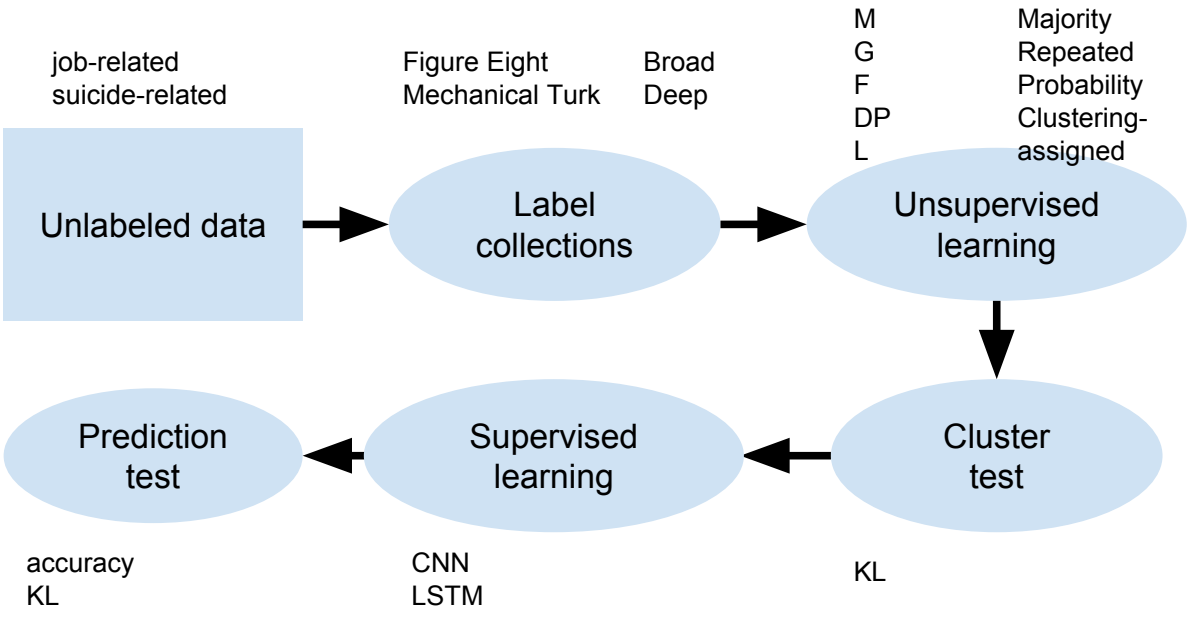

Figure 4.3: Our experiment workflow involves obtaining crowdsourced labels for raw data (yielding empirical label distributions for each data item), trying various unsupervised strategies for aggregating those labels, and finally testing how each approach affects the efficiency of supervised learning prediction. Note there are two testing phases: one for how well each aggregation strategy fits the data and one for supervised learning performance. We also list key terms, keywords, and abbreviations associated with each phase of the workflow.

4.4.1 Clustering Experiments for Ground Truth Estimation

Model Selection

For those clustering models requiring p as a hyperparameter, we test values for $p \in [d/2, 2d]$, where d is the number of label choices. As the estimators for these models are stochastic and/or sensitive to initial conditions and parameters, for every model and every set of hyperparameters we ran 100 trials on the training/dev set and picked the model with the highest estimated likelihood.

We adopt the native likelihood function as our model selection criterion because it is the native optimization goal of each unsupervised clustering algorithm, and provides a consistent strategy for evaluating different unsupervised learning algorithms.

Table 4.2 shows the number of clusters selected on each of the two training splits on each label set and for **DP** the number of clusters the algorithm generated automatically.

	Broad split					**Deep split**				
Dataset	**M**	**G**	**L**	**F**	**DP**	**M**	**G**	**L**	**F**	**DP**
jobQ1FE	10	4	9	3	4	11	11	9	3	4
jobQ1MT	11	4	11	8	10	2	2	11	9	11
jobQ1BOTH	11	2	2	6	8	2	2	11	7	8
jobQ2FE	11	3	10	3	4	11	2	10	3	4
jobQ2MT	2	4	11	7	9	2	2	11	7	10
jobQ2BOTH	2	2	11	5	7	2	2	8	5	7
jobQ3FE	19	5	18	6	7	19	10	19	7	7
jobQ3MT	5	5	14	17	20	5	19	15	17	26
jobQ3BOTH	5	15	18	13	16	5	17	11	17	17
Suicide	8	2	7	4	5	-	-	-	-	-

Table 4.2: The optimal label aggregation models on each label set using two splits (*Broad* and *Deep*) are achieved with the presented number of clusters (p).

Evaluation

Label-based For the model $\underline{M}$ produced by each unsupervised learning algorithm and each data item i in the test set, we determine the most likely cluster j for i's empirical label distribution

ϕ_j: $\arg\max_j P(\hat{y}_i \sim \phi_j \mid \underline{M})$. We then compute the KL divergence between the empirical label distribution $\hat{y}_i$ and the cluster distribution ϕ_j.

Table 4.3 shows that the multinomial mixture models (**M**/**F**/**DP**) generally outperformed **G**, as we expected. The crowdsourced sample sizes of 5–10 labels we used for each training item are typical of crowdsourced supervised learning label sets, and the differences between **G** and the other cluster models appear to be substantial at this scale. The success of **L** on a number of label sets surprised us, considering that we only use the mostly likely cluster for each data item which was trained on a mixture of clusters. Finally, **F** outperforms the other models on all of the sets having at least ten annotations per item, and shows the most improvement from the FE/MT (which had five annotations per item) to the BOTH (with ten annotations per item) label sets.

	Broad split					**Deep split**				
KL	**M**	**G**	**L**	**F**	**DP**	**M**	**G**	**L**	**F**	**DP**
jobQ1FE	0.35	0.53	0.23	0.39	0.39	0.30	0.57	0.24	0.37	0.39
jobQ1MT	0.19	0.68	0.18	0.13	0.15	0.20	0.39	0.07	0.09	0.10
jobQ1BOTH	0.20	0.46	0.40	0.19	0.19	0.21	0.38	0.06	0.06	0.07
jobQ2FE	0.26	0.54	0.19	0.32	0.32	0.24	0.65	0.20	0.28	0.28
jobQ2MT	0.36	0.74	0.15	0.10	0.10	0.26	0.50	0.09	0.11	0.13
jobQ2BOTH	0.28	0.51	0.17	0.16	0.16	0.25	0.48	0.09	0.08	0.08
jobQ3FE	0.51	1.00	0.52	0.59	0.64	0.29	0.97	0.27	0.41	0.41
jobQ3MT	0.50	1.15	0.33	0.26	0.29	0.20	0.51	0.17	0.28	0.21
jobQ3BOTH	0.45	0.82	0.35	0.32	0.33	0.18	0.64	0.18	0.12	0.13
Suicide	0.22	0.57	0.20	0.22	0.22	-	-	-	-	-
Average	0.29	0.59	0.28	0.22	0.23	0.21	0.50	0.11	0.09	0.09
Std dev	0.10	0.14	0.10	0.06	0.06	0.03	0.11	0.05	0.02	0.03

Table 4.3: KL divergence based on the chosen label clustering models in Table 4.2. Average and standard deviation are based on the KL divergence scores of the gray-highlighted rows (jobQ1BOTH, jobQ2BOTH, jobQ3BOTH and Suicide). The *lowest* KL is highlighted in yellow for each split.

Table 4.3 also shows the average and standard deviation of the KL divergence scores on the four independent label sets (i.e., BOTH comprises FE and MT) jobQ1BOTH, jobQ2BOTH, jobQ3BOTH and Suicide (highlighted in gray). These statistics indicate that **F** outperforms the other models across different thematic label sets in its capability and stability, **DP** is second, and, **G** comes last as we expected.

Q3 differs from Q1 and Q2 in allowing annotators to choose more than one label for each tweet. Ideally, the ideal representation for the annotations (where each *annotation* is the set of labels provided by one annotator for one data item) of Q3 would be over the *power set* of possible choices of labels. However, Table 4.4 shows that fewer than 10% of the annotations we received had selected more than one label. To simplify our analysis, we thus treat multiple labels from the same annotator as if each came from its own, independent annotator (for example, an annotation with three labels provided is treated independently as three separate annotations.).

	#labels/worker/item				
Label Set	**1**	**2**	**3**	**4**	**5+**
jobQ3FE	10,000	722	176	53	16
jobQ3MT	12,202	628	58	11	1
jobQ3MT+	2,969	193	32	2	0

Table 4.4: Counts of worker-item pairs, grouped by #labels per worker per data item.

Text-based In addition to the experiments using the unsupervised learning algorithms introduced in Section 4.2.1 to cluster over empirical labels, we additionally clustered on bag-of-word representations of each data item's text and evaluate our label clustering and aggregation models.

We have proposed a new metric to measure probability distributions (in addition to Kullback-Leibler divergence): *entropy gap* (**EG**), calculated as $H(y_i)/\log d - H(z_i)/\log p$, can apply to any label aggregation model or clustering approach that has likelihoods associated with each (data point, cluster) pair and where each point can be interpreted as a probability distribution. The danger with this score is that it is easy to "cheat" to get a good score, say, by assigning all data items to the same cluster.

Tables 4.5 and 4.6 report the EG and KL results, respectively, when we compared the label distributions obtained from the optimal label aggregation models (reported in Table 4.2) against the ones clustered by texts.

4.4.2 Supervised Learning Experiments

We then trained the two supervised learning algorithms described in Section 4.2.2 on our training datasets' *texts*, using in turn each of the unsupervised learning methods described previously to provide *refined label distribution estimates* ($\hat{\mathbf{y}}'_i$) as the learning goal. We compared their performances

	Broad split					Deep split				
EG	**M**	**L**	**G**	**F**	**DP**	**M**	**L**	**G**	**F**	**DP**
jobQ1FE	-0.13	-0.05	0.26	0.03	0.04	-0.08	0.06	0.37	-0.08	0.14
jobQ1MT	0.13	-0.03	0.30	0.03	0.07	0.36	-0.08	0.39	0.36	0.37
jobQ1BOTH	0.13	-0.04	0.23	0.11	0.16	0.32	0.02	0.37	0.34	0.36
jobQ2FE	-0.03	-0.02	0.30	0.04	0.09	0.03	0.06	0.38	0.04	0.12
jobQ2MT	0.26	-0.02	0.30	0.07	0.10	0.39	-0.04	0.39	0.38	0.37
jobQ2BOTH	0.24	0.05	0.29	0.08	0.14	0.39	0.12	0.39	0.33	0.36
jobQ3FE	-0.10	-0.10	0.28	-0.06	0.05	0.01	0.00	0.39	0.03	0.09
jobQ3MT	0.22	-0.05	0.30	0.11	0.14	0.37	0.04	0.38	0.36	0.38
jobQ3BOTH	0.21	-0.01	0.29	0.13	0.16	0.38	0.11	0.39	0.37	0.36
RWsuicide	0.11	0.03	0.33	0.15	0.16	-	-	-	-	-
Average	0.17	0.01	0.29	0.12	0.16	0.36	0.03	0.38	0.35	0.36
Std dev	0.06	0.04	0.04	0.03	0.01	0.04	0.08	0.01	0.02	0.00

Table 4.5: Entropy gap obtained between the optimal label aggregation model and text-based clustering on each dataset using two splits. "EG": Normalized entropy gap (i.e., the average entropy gap per data item). Average and standard deviation are based on the EG scores of the gray-highlighted rows (jobQ1BOTH, jobQ2BOTH, jobQ3BOTH and Suicide). The *lowest* EG is highlighted in yellow for each split.

to those of three common baseline strategies for resolving (or not) label disagreement.

- Majority (**Maj**) takes the final label to be $\hat{y}'_i = \underset{j \in \{1,\ldots,d\}}{\arg\max}\{\hat{y}_{ij}\}$.
- Repeated (**Rept**) duplicates each data instance once for every annotation it receives and pairs the replicated instance with that label.
- Probability (**Prob**) is the raw label distribution estimates $(\hat{\mathbf{y}}'_i) = (\hat{\mathbf{y}}_i)$. (This is the baseline LDL approach.)

Evaluation

We measure the **KL divergence** between the classifier $(\tilde{\mathbf{y}}_i)$ and cluster-or-baseline-method $(\hat{\mathbf{y}}'_i)$ -based label distributions. (Note that Maj and Rept both associate each data item, by eliminating

	Broad split					Deep split				
KL	**M**	**L**	**G**	**F**	**DP**	**M**	**L**	**G**	**F**	**DP**
jobQ1FE	4.72	4.46	4.60	4.84	4.83	2.89	2.39	2.11	3.05	3.10
jobQ1MT	4.50	4.60	4.59	4.50	4.52	3.12	2.70	3.36	2.81	2.83
jobQ1BOTH	4.57	4.78	4.76	4.65	4.65	3.11	2.51	3.12	2.56	2.57
jobQ2FE	4.60	4.40	4.74	4.77	4.77	2.80	2.38	3.20	2.99	2.99
jobQ2MT	4.75	4.60	4.56	4.48	4.48	3.10	2.72	3.33	2.59	2.44
jobQ2BOTH	4.76	4.51	4.83	4.60	4.61	3.10	2.70	3.23	2.49	2.57
jobQ3FE	4.30	4.23	4.54	4.51	4.58	2.50	2.33	2.61	2.65	2.70
jobQ3MT	4.46	4.31	4.49	4.13	4.19	2.62	2.56	2.21	2.26	2.20
jobQ3BOTH	4.44	4.28	4.29	4.18	4.26	2.61	2.64	2.16	2.21	2.22
RWsuicide	4.45	4.42	5.00	4.60	4.61	-	-	-	-	-
Average	4.56	4.50	4.72	4.51	4.53	2.94	2.62	2.84	2.42	2.45
Std dev	0.15	0.21	0.30	0.22	0.18	0.29	0.10	0.59	0.19	0.20

Table 4.6: KL divergence obtained between the optimal label aggregation model and text-based clustering on each dataset using two splits. Average and standard deviation are based on the EG scores of the gray-highlighted rows (jobQ1BOTH, jobQ2BOTH, jobQ3BOTH and Suicide). The *lowest* KL is highlighted in yellow for each split.

labels or creating copies of the data items with exactly one label. For the purpose of computing KL divergence we regard this as a distribution where the entire probability mass is on one label.)

We also measure **Accuracy**, i.e., the percentage of times $\arg\max_{\mathbf{j}} \tilde{\mathbf{y}}_{\mathbf{ij}}$ matches $\arg\max_{\mathbf{j}} \hat{\mathbf{y}}'_{\mathbf{ij}}$ in the test set. Accuracy is often used in nondistributional classification problems. We use it here to shed further light into the differences between distributional and nondistributional problems. In particular, we might expect that nondistributional models might outperform label distribution models with respect to accuracy, even as they underperform with respect to KL divergence.

Results

Tables 4.7 and 4.8 show the KL divergence and accuracy metrics for CNN/LSTM text classifiers built with different label aggregation strategies in two split modes (Broad and Deep).

Starting with the KL divergence results, on the Broad split tests, CNNs trained and tested on **L** outperform other clustering and non-clustering approaches most of the time for both job and

CNN	**KL divergence**								**Accuracy**							
	Maj	Rept	Prob	M	G	L	F	DP	Maj	Rept	Prob	M	G	L	F	DP
jobQ1FE	2.98	0.79	0.91	0.12	0.74	0.47	0.18	0.19	0.73	0.53	0.72	0.78	0.95	0.58	0.64	0.58
jobQ1MT	2.03	0.80	0.72	0.65	1.05	0.52	1.02	1.00	0.80	0.72	0.79	0.56	0.67	0.76	0.54	0.56
jobQ1BOTH	2.38	0.45	0.48	0.36	0.38	0.27	0.40	0.38	0.82	0.64	0.81	0.57	0.76	0.76	0.65	0.64
jobQ2FE	2.29	0.91	0.79	0.21	0.78	0.13	0.31	0.28	0.73	0.63	0.79	0.71	0.62	0.94	0.59	0.64
jobQ2MT	2.10	0.80	0.78	0.81	0.98	0.67	1.04	0.96	0.73	0.68	0.73	0.48	0.55	0.71	0.53	0.52
jobQ2BOTH	2.12	0.49	0.47	0.48	0.48	0.37	0.51	0.52	0.76	0.65	0.76	0.63	0.58	0.71	0.54	0.56
jobQ3FE	4.20	1.66	1.14	0.31	0.68	0.66	0.42	0.36	0.36	0.31	0.41	0.47	0.32	0.45	0.42	0.46
jobQ3MT	3.18	2.24	1.05	1.04	1.32	0.54	1.12	1.12	0.53	0.45	0.51	0.26	0.28	0.49	0.28	0.28
jobQ3BOTH	3.38	1.40	0.77	0.62	0.49	0.62	0.71	0.70	0.48	0.42	0.53	0.31	0.62	0.46	0.25	0.21
Suicide	2.16	1.40	0.45	0.69	13.62	0.33	0.53	0.49	0.81	0.65	0.78	0.18	1.00	0.76	0.37	0.39
Average	2.51	0.94	0.54	0.54	3.74	0.40	0.54	0.52	0.72	0.59	0.72	0.42	0.74	0.67	0.45	0.45
Std dev	0.51	0.47	0.13	0.13	5.70	0.13	0.11	0.11	0.14	0.10	0.11	0.18	0.16	0.12	0.15	0.17
LSTM	Maj	Rept	Prob	M	G	L	F	DP	Maj	Rept	Prob	M	G	L	F	DP
jobQ1FE	0.80	0.91	1.12	0.49	0.66	0.63	0.33	0.53	0.84	0.75	0.87	0.89	0.99	0.76	0.83	0.84
jobQ1MT	1.09	1.16	1.15	1.37	1.49	1.40	1.07	0.62	0.86	0.82	0.85	0.81	0.87	0.83	0.80	0.81
jobQ1BOTH	0.75	0.80	0.54	1.12	0.45	0.83	0.52	1.09	0.88	0.79	0.86	0.81	0.89	0.82	0.82	0.82
jobQ2FE	1.25	1.12	1.20	0.79	0.94	1.14	1.14	0.54	0.85	0.78	0.87	0.85	0.82	0.97	0.84	0.82
jobQ2MT	1.88	1.07	1.48	0.92	1.52	1.24	1.71	1.25	0.84	0.81	0.82	0.78	0.83	0.81	0.79	0.80
jobQ2BOTH	0.86	0.78	1.68	1.50	0.67	0.93	0.89	0.73	0.86	0.80	0.84	0.86	0.81	0.81	0.80	0.83
jobQ3FE	1.79	1.68	1.54	0.93	1.13	1.14	1.01	0.92	0.64	0.64	0.62	0.65	0.72	0.71	0.65	0.62
jobQ3MT	2.08	1.65	1.86	1.81	1.73	1.18	1.88	1.58	0.69	0.70	0.63	0.63	0.59	0.50	0.66	0.67
jobQ3BOTH	1.26	1.46	1.99	1.46	1.10	1.38	1.42	1.40	0.70	0.68	0.63	0.63	0.86	0.61	0.61	0.66
Suicide	1.14	0.74	0.85	0.67	13.97	0.91	0.68	0.85	0.74	0.72	0.74	0.73	0.50	0.72	0.71	0.71
Average	1.00	0.95	1.27	1.19	4.05	1.01	0.88	1.02	0.80	0.75	0.77	0.76	0.77	0.74	0.74	0.76
Std dev	0.21	0.30	0.59	0.33	5.73	0.22	0.34	0.26	0.08	0.05	0.09	0.09	0.16	0.08	0.08	0.07

Table 4.7: KL divergence and accuracy of the **Broad** split. Average and standard deviation are based on the gray-highlighted rows (jobQ1BOTH, jobQ2BOTH, jobQ3BOTH and Suicide). The *lowest* KL and *highest* accuracy are highlighted in yellow.

CNN	**KL divergence**								**Accuracy**							
	Maj	Rept	Prob	M	G	L	F	DP	Maj	Rept	Prob	M	G	L	F	DP
jobQ1FE	3.09	0.77	0.90	0.13	0.69	0.39	0.09	0.16	0.62	0.47	0.58	0.78	0.80	0.54	0.82	0.72
jobQ1MT	2.94	0.47	0.54	0.64	1.08	0.47	1.22	1.05	0.72	0.53	0.70	0.58	0.66	0.72	0.56	0.58
jobQ1BOTH	2.90	0.34	0.24	0.39	0.43	0.38	0.33	0.35	0.72	0.51	0.70	0.62	0.90	0.60	0.62	0.60
jobQ2FE	3.07	0.57	0.65	0.18	0.56	0.49	0.21	0.31	0.60	0.53	0.52	0.76	0.82	0.48	0.50	0.60
jobQ2MT	1.90	0.50	0.58	0.77	0.68	0.76	0.74	1.07	0.72	0.57	0.70	0.58	0.64	0.66	0.64	0.44
jobQ2BOTH	2.90	0.27	0.28	0.52	0.37	0.35	0.50	0.58	0.72	0.54	0.76	0.54	0.56	0.68	0.64	0.54
jobQ3FE	3.71	1.45	1.00	0.34	0.63	0.65	0.53	0.43	0.46	0.40	0.48	0.16	0.30	0.50	0.14	0.40
jobQ3MT	3.95	1.98	0.77	1.13	1.21	1.20	1.26	1.24	0.54	0.43	0.54	0.14	0.24	0.48	0.30	0.18
jobQ3BOTH	3.33	1.13	0.63	0.76	0.67	0.49	0.71	0.73	0.62	0.46	0.48	0.20	0.40	0.56	0.16	0.24
Average	3.04	0.58	0.38	0.56	0.49	0.41	0.51	0.55	0.69	0.50	0.65	0.45	0.62	0.61	0.47	0.46
Std dev	0.20	0.39	0.18	0.15	0.13	0.06	0.16	0.16	0.05	0.03	0.12	0.18	0.21	0.05	0.22	0.16
LSTM	Maj	Rept	Prob	M	G	L	F	DP	Maj	Rept	Prob	M	G	L	F	DP
jobQ1FE	0.94	0.85	0.76	0.52	0.89	0.61	0.79	0.65	0.74	0.70	0.74	0.92	0.93	0.67	0.92	0.91
jobQ1MT	0.78	0.53	0.52	0.88	1.08	1.10	1.80	1.38	0.71	0.71	0.71	0.79	0.87	0.70	0.82	0.82
jobQ1BOTH	0.39	0.65	0.70	0.63	0.72	0.54	0.86	0.64	0.70	0.71	0.70	0.81	0.98	0.69	0.84	0.85
jobQ2FE	0.99	1.16	1.04	1.37	0.57	0.90	0.98	0.88	0.77	0.72	0.77	0.87	0.91	0.75	0.85	0.86
jobQ2MT	0.83	0.94	0.77	0.96	1.44	1.32	1.15	1.13	0.69	0.69	0.69	0.76	0.83	0.66	0.81	0.81
jobQ2BOTH	0.88	0.63	0.73	0.67	1.30	0.94	1.31	1.15	0.69	0.69	0.69	0.86	0.85	0.64	0.80	0.83
jobQ3FE	1.97	1.55	1.40	0.84	0.90	1.71	0.92	1.04	0.62	0.65	0.67	0.66	0.83	0.70	0.64	0.59
jobQ3MT	1.51	1.38	1.44	1.65	2.01	1.99	1.63	1.67	0.68	0.61	0.62	0.64	0.59	0.62	0.67	0.60
jobQ3BOTH	1.41	1.28	1.26	1.43	1.06	1.54	1.29	1.46	0.62	0.61	0.68	0.64	0.71	0.67	0.68	0.58
Average	0.89	0.85	0.90	0.91	1.03	1.01	1.15	1.08	0.67	0.67	0.69	0.77	0.85	0.67	0.77	0.75
Std dev	0.42	0.30	0.26	0.37	0.24	0.41	0.21	0.34	0.04	0.04	0.01	0.09	0.11	0.02	0.07	0.12

Table 4.8: KL divergence and accuracy of the **Deep** split. Average and standard deviation are based on the gray-highlighted rows (jobQ1BOTH, jobQ2BOTH, and jobQ3BOTH). The *lowest* KL and *highest* accuracy are highlighted in yellow.

suicide discourse themes. For LSTMs, we can also observe that clustering approaches achieved better results more often on different label sets than non-clustering methods. Almost none of CNNs or LSTMs trained on any baseline label reduction strategy can compete.

By contrast, the results of the Deep split KL divergence tests (Table 4.8) are not as conclusive, and this could be due to there being fewer data items in the Deep split test set. But even so, clustering strategies again perform better in more cases than the baselines.

Tables 4.7 and 4.8 show that, for both the CNN and LSTM classifiers and both split modes, the highest accuracies often come from the clustering methods. They outperform non-clustering methods by more than 10% on average, which appears substantial. For those label sets whose accuracy based on clustering strategies do not rank 1st, non-clustering methods win only by a slim or zero margins.

Together, the results for different label sets and split modes reveal several interesting patterns. First, the cluster-based models tend to outperform the baseline methods in terms of either KL divergence or accuracy. This supports the feasibility of our clustering strategy for label distribution learning on subjective problems with annotator disagreement. On the other hand, for conventional (i.e., non-distributional) classification problems, baseline methods can be sufficient. The advantages of clustering, in terms of KL divergence, is less stark in the Deep compared to the Broad splits, but clustering still seems to outperform baselines on the jobQ3 label set, which has the largest label space and is where pooling and other label conservation methods are most needed.

4.5 Discussion

Our results provides evidence—both for and against—that clustering is a feasible strategy to improve performance of label distribution learning in certain settings, such as when each label distribution represents a population-level estimate based on a (micro) sample, and the data fall into a small number of semantically-equivalent and similarly-rated classes (relative to the complexity of learning task). Our results shed a little light on the validity of the clustering theory.

They also raise methodological issues. We expect that the methods introduced for testing performance will provide helpful baselines for the development of newer quantitative methods tailored specifically toward settings where ground truth is generated from a small number of sample labels per data item.

In retrospect, when we selected the best label aggregation (clustering) models tested on each dataset, we should have experimented with other alternative criteria. These alternative methods for model selection could be the Akaike information criterion (AIC) [246] and the Bayesian information criterion (BIC) [247], both of which are maximum likelihood estimation driven and can deal with the risks of overfitting and underfitting at the same time.

Another methodological issue we grappled with was whether to measure the performances of the supervised classification models against the empirical ($\hat{y}$) or refined ($\hat{y}'$) label distributions. Common practice is to test supervised learning on the patterns they are fed (i.e., the refined labels in our case). But in our case the conventional machine learning algorithms are only the second half of a longer pipeline that has essentially an unsupervised front end, and which takes the empirical label distributions as input. We tried both approaches, but because we found our results more interesting in this direction, we report on the predictions against $\hat{y}'$. The biggest worry in doing so is that, because pooling labels via a small number of clusters greatly reduces diversity in the label distributions, there is less likelihood of error, which would seem to make predictions artificially easier against $\hat{y}'$ than the empirical distributions $\hat{y}$. On the other hand, since these clusters are based on raw labels, the larger the clusters the greater the likelihood that items with inconsistent features are assigned to the same cluster, and this would lead to less accurate predictions from the supervised models.

One may be curious about some generic, distance-based clustering methods and metrics other than those we applied in the unsupervised learning stage. We have not considered those models yet (the Gaussian model was chosen because though generative it did not support our hypothesis that related labels are all multinomial samples of a latent distribution), mainly because they did not directly relate to our hypothetical framework, and we wanted to take a principled approach, given the vast number of alternative models to consider in the field.

We have been deliberately vague about the meaning of "a population of labelers." This study was motivated by our previous work with microtask crowdsourcing platforms like Amazon Mechanical Turk and Figure Eight, in which case our collected labels can be considered as a collection of (micro) samples of the population of workers on whichever sites are used for whatever interval of time the requested labeling task is posted. Studies exist on the demographics of these sites. Some websites (like Figure Eight) provide some demographic information on the responders to each microtask request.

We have not yet modeled user behavior comprehensively, though this is a well-established approach for aggregating labels from multiple annotators. In fact, we did run experiments using Dawid and

Skene's class annotator-based model [1], which is based mainly on user behavior. However, as it is designed for conventional, non-distributional supervised learning and did not perform well for our experimental settings, we did not report those results. Another complication is that most of our annotators labeled only ten data items each, so we would be tempted to cluster users in much the same way we used clustering here to group data items.

Another limitation was that we did not investigate in-depth the causes of inter-annotator disagreements, such as data encoding errors and communication ambiguities [44, 47, 60], lack of sufficient information [42, 60, 61], and unreliable annotators and their bias [42], nor did we attempt to resolve the disagreement through follow-up discussions with the annotators, as is common in many grounded theory studies.

We suspect in our experiment label sets that there are some statistical correlations between the subjectivity and ambiguity and the degree of inter-rater disagreement across different questions. We hope to explore these directions in the future.

Another potential future direction could be to explore more highly structured label spaces, such as ordinal ones, or ones based on Bernoulli distributions or "single-peaked-ness" that are common in practice and sometimes yield to high-performance algorithms.

4.6 Summary

We study the important problem of predicting the distributions of population beliefs by integrating unsupervised and supervised learning methods. We test different strategies for clustering data items to obtain aggregated label distributions. We then build supervised CNN/LSTM classifiers using the predicted distributions from the clustering models and compared their performance with common baseline label reduction strategies. Our results from both unsupervised and supervised experiments show that it is feasible to predict probability distributions over labels at the population level. Clustering labels, in general, boost the label distribution learning by aggregating data items with similar semantics and population beliefs. We believe our study is a pioneering exploration of disagreement on linguistic data from social media and further helps future intelligent agents understand the diversity of opinions in society and the real world.

Chapter 5

Future Work

When we combine label distribution learning (Chapter 4) with our existing active learning framework (Chapter 3.3.2), it is logical to further explore the feasibility and performance of iteratively learning label distributions for more complicated problems with subjective annotations. We first introduce our proposed work in general, and follow up with more technical details in the process of collecting annotations and building models.

5.1 Active Learning with Humans in the Loop

We propose our advanced humans-in-the-loop active learning framework with query strategies in Algorithm 1. In Figure 5.1, we illustrate our framework for iteratively learning the accurate probabilistic label distributions.

In our proposed framework, we update the classification model progressively with humans in the loop to annotate queried informative instances with subjectivity through multiple iterations, to improve class prediction performance.

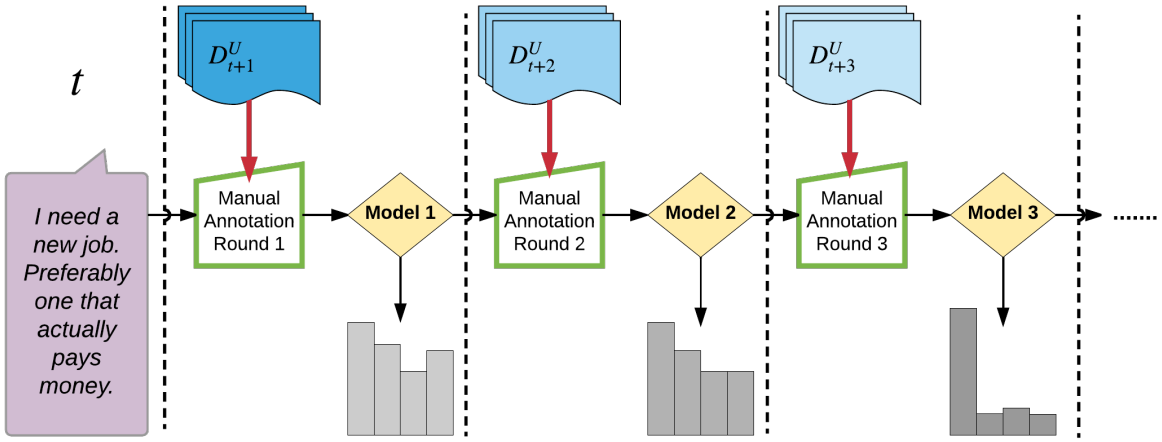

Figure 5.1: Illustration of our humans-in-the-loop active learning framework to progressively learn the accurate distribution of label probabilities. D^U on the top row denotes the unlabeled data pool, which is consumed gradually $(t, t+1, t+2, t+3, \dots)$ (represented as from dark blue to light blue along the time line). The probabilistic distribution of labels of the tweets (bars on the bottom row) in the validation set are updated after each round of model training—bars with darker shade indicates more accurate class probability estimation. At each time step t, we aim to exploit the current model (θ_t) predictions to intelligently query the estimated most informative samples from D^U_t and then train a new model at $t+1$ with new labels collected from human annotators (represented as trapezoids).

Input: A set of initial labeled samples D^L_0, a set of initial unlabeled samples D^U_0, number of new samples to query each time u, number of label queries per sample h, maximum iteration number m

Output: Model θ_t

$t \rightarrow 1$

while $t \leq m$ **do**

  Add $u \times h$ human-annotated samples with labels into D^L_t based on Equation 5.4 and 5.5, respectively;

  Construct θ_t from D^L_t;

  $t \leftarrow t+1$

end

Algorithm 1: Humans-in-the-loop learning framework

5.2 Proposed Details

In this section, we briefly introduce some experiment design with specific parameters and settings which would be considered in our proposed work.

5.2.1 Loss Function

Besides KL divergence (as used in Section 4.2.2), we want to study other alternative performance metrics in our proposed work.

Following the same notation as Section 2.3.2, let $p^{\theta}(y|x)$ denote the probability that a data item $x \in \mathcal{U}$ is labeled as $y \in \{1, \cdots, m\}$ according to model θ, where for the data item x, y is a random variable, and p^{θ} represents the probability distribution according to model θ. There exists a hidden true label distribution $\mathbb{P}(x)$. The best we can do is to collect labeling samples from multiple annotators to estimate $\mathbb{P}(x)$. Let $\hat{f}(x, D^L)$ denote such an estimate for x based on a labeled dataset D^L.

Jensen-Shannon divergence (JSD) can be deployed to measure the symmetric similarity among two or more probability distributions—it quantifies how "distinguishable" a set of probability distributions are from another. For every sample $x \in \mathcal{U}$, we measure, at every time step t, the similarity between the model predicted probability distribution $p^{\theta_t}(y|x)$ and the estimated probability distribution $\hat{f}(x, D_t^L)$. In contrast, in most conventional classification problems involving neural models, the loss is measured by computing the cross-entropy error between the softmax of the output and the target (true) label distribution. We do not use cross-entropy here because it penalizes for higher entropy output while our active learning cycles should ideally be invariant to the entropy of the input samples.

Similarly, we formulate our learning task in subjective domains using JSD as the problem of maximizing the following objective function:

$$\arg\min_{\theta_t} |\mathcal{U}| \sum_{x \in \mathcal{U}} JSD(p^{\theta_t}(y|x), \hat{f}(x, D_t^L)) \tag{5.1}$$

Equation 5.1 represents a general optimization object and should be applied to any base learner which is denoted as θ.

5.2.2 Query Strategies

Common active learning query strategies do not apply in subjective domains with probabilistic distributions because they are based on the assumption that the only legitimate distribution of labels is that where exactly one class label is non-zero or, equivalently, with zero entropy.

On the contrary, we are as interested in data with high- as with as low- entropy label distributions. Further, we would like to study the modeling of those probabilistic distributions in all phases of the learning cycle.

Our clustering theory basically hypothesizes that in many settings the causes of subjectivity are limited, and this, in turn, limits the potential number of distinct label distributions present over all elements in the data. We would thus expect good estimates of the true label distributions to cluster around a relatively small number of points in label space.

We will empirically test whether we can improve the classification performance of an active learning algorithm via query strategies that account for this hypothesis in their design. We will, for instance, compare the performance of the traditional least confidence sampling to the following proposed query strategy, which can be seen as a nonparametric strategy.

Nonparametric Choose at time t the item to be manually labeled as

$$\arg\max_{x \in D_t^U} \{ \min_{v \in D_t^U} \{ |p^{\theta_t}(y|x) - p^{\theta_t}(y|v)| \} \}, \tag{5.2}$$

where $|p^{\theta_t}(y|x) - p^{\theta_t}(y|v)|$ is the **cosine similarity** between $p^{\theta_t}(y|x)$ and $p^{\theta_t}(y|v)$. The idea here is straightforward: the item whose label distribution is estimated to be furthest away from all the others in the training set should be theoretically the furthest away from any potential cluster centers, and thus its current estimate is in the most need of updating next at $t + 1$.

Beyond cosine similarity, we may explore clustering algorithms that we have not experimented with (such as k-means clustering) to group labels into a small number of classes and then query the item x that is the least likely to be in any of the classes $C \in \mathcal{C}$.

$$\arg\min_{x \in D_t^U} \{ \max_{C \in \mathcal{C}} \{ \mathcal{L}(x \in C) \} \}, \tag{5.3}$$

where the likelihood function $\mathcal{L}$ depends on the specific clustering algorithm.

Large JSD Sampling This idea can be seen as a variant of uncertainty sampling, which is a commonly-used query strategy in traditional active learning problems. We will similarly select some samples that could be the most informative and representative according to the uncertainty measures (which are confidence, margin sampling and entropy as introduced in Section 2.3.2). Such uncertainty-like sampling methods can provide a good coverage of outliers of model predictions.

With this kind of measure in place, one simple heuristic—similar to choosing the most uncertain samples in normal active learning problems—is to query the data item x_i from D_t^U under the model θ whose estimated probability distribution p_i^θ is the furthest from the average estimate of the rest samples (the data item x_i excluded) $\bar{p_i^\theta}$:

$$\arg\max_i(JSD(p_i^\theta, \bar{p_i^\theta})). \tag{5.4}$$

The above idea (Equation 5.4) may apply to cosine similarity—another measure of distance between distributions—instead of JSD.

Small JSD Sampling Besides querying outlier samples, we further propose a model-based query strategy to sample data with small JSD which covers a majority of certain unlabeled samples. This query strategy is useful for models to consistently learn good feature representations and beneficial to improve the accuracy and stability of models over multiple iterations in active learning settings.

The samples' pseudo label distributions are assigned by the current model θ and we raise queries as the following representations:

$$\arg\min_i(JSD(p_i^\theta, \bar{p_i^\theta})). \tag{5.5}$$

5.2.3 Training Convergence

In our active learning framework, we repeat the above query steps to update the labeling and training process, until we reach the stopping criteria, such as the maximum rounds of iterations m or acceptable performance output for the model (e.g., the machine predicted label distributions of test set are approximate to human label distributions).

One learning measure is to calculate the average JSD over multiple rounds of training to determine

the stopping criteria and investigate the model learning convergence. When a classifier (θ) provides a distribution of class probabilities (p^{θ}) for a given example (x), JSD can be applied to compute a measure of similarities between the distributions produced by a series of such classifiers (θ_t, $t \in \{1, \cdots, T\}$) as:

$$JSD(p^{\theta_1}, \cdots, p^{\theta_T} | f(x)) = H(\sum_{t=1}^{T} p^{\theta_t}) - \sum_{t=1}^{T} H(p^{\theta_t}), \tag{5.6}$$

where $H(p)$ is the Shannon entropy for the probability distribution p, defined as:

$$H(p^{\theta}) = -\sum_{j=1}^{m} \Pr_j^{\theta} \log \Pr_j^{\theta} \tag{5.7}$$

where $\Pr_j^{\theta}$ represents the probability that the model θ assigns any single class label $j \in \{1, \cdots, m\}$ (to the example x).

Chapter 6

Conclusion

In this thesis, we focused on learning population label distributions from crowdsourced annotations with humans in the loop to understand social media data. The biggest motivation comes from the subjectivity issues embedded in social media data and classification problems where no authoritative or gold standard data exist.

In Chapter 2, we reviewed crowdsourcing techniques and humans-in-the-loop learning that integrates human contributors with machine learning in an effective, efficient and inexpensive way. We reviewed multi-label learning techniques which account for data items associated with subjective diverse judgments, and active learning techniques that can achieve better modeling outcomes with less training data. We also summarized a series of previous NLP research and applications which are relevant to our thesis study, including text categorization, social issues in social media, semantic frames, and neural network modeling, and so on.

In Chapter 3, we reviewed our previous research which laid foundations for our thesis work. We demonstrated that non-expert and expert annotators performed differently and that labels produced by each community led to different modeling performances and reflected differences among each group in understanding social issues like distress and suicide. We designed crowdsourced annotation tasks and showed in a series of experiments that it is useful to build robust machine models using unanimous votes from multiple annotators. We developed a humans-in-the-loop active learning framework that integrates crowdsourcing contributions, local community knowledge, and linguistic features to identify job-related tweets from individual and business accounts on public social media, and contributed the Twitter Job/Employment Corpus to the research community.

In Chapter 4, we formalized a population label distribution learning problem with a *clustering theory* and introduced our LDL algorithms to estimate ground truth and predict label distributions from crowdsourced annotations with disagreement. We designed annotation schemes to collect multiple annotations from crowdsourced workers. We conducted comparative experiments to comprehensively evaluate different labeling strategies, clustering algorithms and neural network classification models. We discussed model selection criteria and evaluation metrics in probabilistic learning scenario. From the dataset perspective, by observing the various employment stages expressed in the job-related tweets posted by individuals, we explored them as an issue with subjectivity in this thesis work, together with suicide as another representative subjective domain.

In Chapter 5, we proposed our future research direction based on our existing efforts, to tackle more challenging problems by integrating active learning with humans in the loop.

Bibliography

[1] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Applied statistics*, pp. 20–28, 1979.

[2] P. Welinder and P. Perona, "Online crowdsourcing: rating annotators and obtaining cost-effective labels," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pp. 25–32, IEEE, 2010.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[5] C. Homan, R. Johar, T. Liu, M. Lytle, V. Silenzio, and C. Ovesdotter Alm, "Toward macro-insights for suicide prevention: Analyzing fine-grained distress at scale," in *Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, (Baltimore, Maryland, USA), pp. 107–117, Association for Computational Linguistics, June 2014.

[6] T. Liu, Q. Cheng, C. Homan, and V. Silenzio, "Learning from various labeling strategies for suicide-related messages on social media: An experimental study.," in *The workshop on Mining Online Health Reports of the 10th ACM Conference on Web Search and Data Mining*, (Cambridge, UK), February 2017.

[7] T. Liu, C. M. Homan, C. O. Alm, A. M. White, M. C. Lytle, and H. A. Kautz, "Understanding discourse on work and job-related well-being in public social media," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 1044–1053, Association for Computational Linguistics, August 2016.

[8] A. Sadilek, H. Kautz, and V. Silenzio, "Predicting disease transmission from geo-tagged micro-blog data," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pp. 136–142, AAAI Press, 2012.

[9] Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: Liwc and computerized text analysis methods," *Journal of language and social psychology*, vol. 29, no. 1, pp. 24–54, 2010.

[10] J. Jashinsky, S. H. Burton, C. L. Hanson, J. West, C. Giraud-Carrier, M. D. Barnes, and T. Argyle, "Tracking suicide risk factors through Twitter in the US.," *Crisis*, vol. 35, no. 1, pp. 51–59, 2014.

[11] D. Altman, "Inter-rater agreement," *Practical statistics for medical research*, vol. 5, pp. 403–409, 1991.

[12] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[13] R. G. Pontius, O. Thontteh, and H. Chen, "Components of information for multiple resolution comparison between maps that share a real variable," *Environmental and Ecological Statistics*, vol. 15, no. 2, pp. 111–142, 2008.

[14] C. J. Willmott and K. Matsuura, "On the use of dimensioned measures of error to evaluate the performance of spatial interpolators," *International Journal of Geographical Information Science*, vol. 20, no. 1, pp. 89–102, 2006.

[15] D. Lazer, A. S. Pentland, L. Adamic, S. Aral, A. L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, *et al.*, "Life in the network: the coming age of computational social science," *Science (New York, NY)*, vol. 323, no. 5915, p. 721, 2009.

[16] E. Asano, "How much time do people spend on social media?," 2017.

[17] J. Li, A. Ritter, C. Cardie, and E. H. Hovy, "Major life event extraction from twitter based on congratulations/condolences speech acts.," in *EMNLP*, pp. 1997–2007, 2014.

[18] J. Pearson, *Why An AI-Judged Beauty Contest Picked Nearly All White Winners*, 2016. (accessed November 11, 2018).

[19] The Guardian, "Microsoft 'deeply sorry' for racist and sexist tweets by ai chatbot," 2016. (Accessed November 11, 2018).

[20] Y. Ren and X. Geng, "Sense beauty by label distribution learning," in *Proc, IJCAI*, pp. 2648–2654, 2017.

[21] X. Geng, "Label distribution learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1734–1748, 2016.

[22] X. Geng and P. Hou, "Pre-release prediction of crowd opinion on movies by label distribution learning.," in *IJCAI*, pp. 3511–3517, 2015.

[23] X. Geng, Q. Wang, and Y. Xia, "Facial age estimation by adaptive label distribution learning," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pp. 4465–4470, IEEE, 2014.

[24] R. C. Solomon, "Subjectivity," in *The Oxford Companion to Philosophy* (T. Honderich, ed.), p. 900, Oxford University Press, 2005.

[25] C. O. Alm, "Subjective natural language problems: Motivations, applications, characterizations, and implications," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pp. 107–112, Association for Computational Linguistics, 2011.

[26] D. P. W. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence, "The quest for ground truth in musical artist similarity.," 01 2002.

[27] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie, "Visual recognition with humans in the loop," in *European Conference on Computer Vision*, pp. 438–451, Springer, 2010.

[28] M.-L. Zhang, "A k-nearest neighbor based multi-instance multi-label learning algorithm," in *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, vol. 2, pp. 207–212, IEEE, 2010.

[29] D. C. Brabham, *Crowdsourcing*. Mit Press, 2013.

[30] L. von Ahn and L. Dabbish, "Labeling images with a computer game," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, (New York, NY, USA), pp. 319–326, ACM, 2004.

[31] L. von Ahn, M. Kedia, and M. Blum, "Verbosity: A game for collecting common-sense facts," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, (New York, NY, USA), pp. 75–78, ACM, 2006.

[32] Wikipedia contributors, "Crowdsourcing — Wikipedia, the free encyclopedia," 2019. [Online; accessed 2-December-2019].

[33] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks," in *Proceedings of the conference on empirical methods in natural language processing*, pp. 254–263, Association for Computational Linguistics, 2008.

[34] C. Callison-Burch, "Fast, cheap, and creative: Evaluating translation quality using amazon's mechanical turk," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, (Stroudsburg, PA, USA), pp. 286–295, Association for Computational Linguistics, 2009.

[35] M. Denkowski and A. Lavie, "Exploring normalization techniques for human judgments of machine translation adequacy collected using amazon mechanical turk," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, (Stroudsburg, PA, USA), pp. 57–61, Association for Computational Linguistics, 2010.

[36] M. Schulze, "A new monotonic and clone-independent single-winner election method," *Voting matters*, vol. 17, no. 1, pp. 9–19, 2003.

[37] K. Evanini, D. Higgins, and K. Zechner, "Using amazon mechanical turk for transcription of non-native speech," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, (Stroudsburg, PA, USA), pp. 53–56, Association for Computational Linguistics, 2010.

[38] M. Marge, S. Banerjee, and A. I. Rudnicky, "Using the amazon mechanical turk to transcribe and annotate meeting speech for extractive summarization," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, (Stroudsburg, PA, USA), pp. 99–107, Association for Computational Linguistics, 2010.

[39] L. B. Chilton, G. Little, D. Edge, D. S. Weld, and J. A. Landay, "Cascade: Crowdsourcing taxonomy creation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, (New York, NY, USA), pp. 1999–2008, ACM, 2013.

[40] C. Akkaya, A. Conrad, J. Wiebe, and R. Mihalcea, "Amazon mechanical turk for subjectivity word sense disambiguation," in *Proceedings of the NAACL HLT 2010 Workshop on Creating*

Speech and Language Data with Amazon's Mechanical Turk, CSLDAMT '10, (Stroudsburg, PA, USA), pp. 195–203, Association for Computational Linguistics, 2010.

[41] G. Parent and M. Eskenazi, "Clustering dictionary definitions using amazon mechanical turk," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, (Stroudsburg, PA, USA), pp. 21–29, Association for Computational Linguistics, 2010.

[42] R. J. Hickey, "Noise modelling and evaluating learning from examples," *Artificial Intelligence*, vol. 82, no. 1, pp. 157–179, 1996.

[43] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[44] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artificial intelligence review*, vol. 22, no. 3, pp. 177–210, 2004.

[45] J. A. Sáez, M. Galar, J. Luengo, and F. Herrera, "Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition," *Knowledge and information systems*, vol. 38, no. 1, pp. 179–206, 2014.

[46] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2014.

[47] D. Angluin and P. Laird, "Learning from noisy examples," *Machine Learning*, vol. 2, no. 4, pp. 343–370, 1988.

[48] B. Edmonds, "The nature of noise," in *Epistemological Aspects of Computer Simulation in the Social Sciences*, pp. 169–182, Springer, 2009.

[49] D. M. Hawkins, *Identification of outliers*, vol. 11. Springer, 1980.

[50] R. J. Beckman and R. D. Cook, "Outlier. s," *Technometrics*, vol. 25, no. 2, pp. 119–149, 1983.

[51] V. Barnett and T. Lewis, *Outliers in statistical data*. Wiley, 1974.

[52] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.

[53] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *Advances in neural information processing systems*, pp. 582–588, 2000.

[54] P. Hayton, B. Schölkopf, L. Tarassenko, and P. Anuzis, "Support vector novelty detection applied to jet engine vibration spectra," in *NIPS*, pp. 946–952, 2000.

[55] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[56] H. Hoffmann, "Kernel pca for novelty detection," *Pattern Recognition*, vol. 40, no. 3, pp. 863–874, 2007.

[57] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[58] D. Collett and T. Lewis, "The subjective nature of outlier rejection procedures," *Applied Statistics*, pp. 228–237, 1976.

[59] X. Liu, G. Cheng, and J. X. Wu, "Analyzing outliers cautiously," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 432–437, 2002.

[60] C. E. Brodley and M. A. Friedl, "Identifying mislabeled training data," *Journal of artificial intelligence research*, vol. 11, pp. 131–167, 1999.

[61] P. Brazdil and P. Clark, "Learning from imperfect data," in *Machine Learning, Meta-Reasoning and Logics*, pp. 207–232, Springer, 1990.

[62] A. Malossini, E. Blanzieri, and R. T. Ng, "Detecting potential labeling errors in microarrays by data perturbation," *Bioinformatics*, vol. 22, no. 17, pp. 2114–2121, 2006.

[63] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi, "Inferring ground truth from subjective labelling of venus images," *Advances in neural information processing systems*, vol. 7, pp. 1085–1092, 1995.

[64] P. Smyth, "Bounds on the mean classification error rate of multiple experts," *Pattern Recognition Letters*, vol. 17, no. 12, pp. 1253–1257, 1996.

[65] N. P. Hughes, S. J. Roberts, and L. Tarassenko, "Semi-supervised learning of probabilistic models for ecg segmentation," in *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, vol. 1, pp. 434–437, IEEE, 2004.

[66] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: The penn treebank," *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[67] C. E. Brodley, M. A. Friedl, *et al.*, "Identifying and eliminating mislabeled training instances," in *AAAI/IAAI, Vol. 1*, pp. 799–805, 1996.

[68] L. Joseph, T. W. Gyorkos, and L. Coupal, "Bayesian estimation of disease prevalence and the parameters of diagnostic tests in the absence of a gold standard," *American Journal of Epidemiology*, vol. 141, no. 3, pp. 263–272, 1995.

[69] A. Gaba and R. L. Winkler, "Implications of errors in survey data: a bayesian model," *Management Science*, vol. 38, no. 7, pp. 913–925, 1992.

[70] C. J. Pérez, F. J. Girón, J. Martín, M. Ruiz, and C. Rojano, "Misclassified multinomial data: a bayesian approach.," *RACSAM*, vol. 101, no. 1, pp. 71–80, 2007.

[71] E. Eskin, "Detecting errors within a corpus using anomaly detection," in *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pp. 148–153, Association for Computational Linguistics, 2000.

[72] N. D. Lawrence and B. Schölkopf, "Estimating a kernel fisher discriminant in the presence of label noise," in *ICML*, vol. 1, pp. 306–313, Citeseer, 2001.

[73] Y. Li, L. F. Wessels, D. de Ridder, and M. J. Reinders, "Classification in the presence of class noise using a probabilistic kernel fisher method," *Pattern Recognition*, vol. 40, no. 12, pp. 3349–3357, 2007.

[74] J. Bootkrajang and A. Kabán, "Multi-class classification in the presence of labelling errors.," in *ESANN*, pp. 345–350, Citeseer, 2011.

[75] B. Frénay, G. de Lannoy, and M. Verleysen, "Label noise-tolerant hidden markov models for segmentation: application to ecgs," *Machine learning and knowledge discovery in databases*, pp. 455–470, 2011.

[76] C. Bouveyron, S. Girard, and M. Olteanu, "Supervised classification of categorical data with uncertain labels for dna barcoding.," in *ESANN*, 2009.

[77] U. Rebbapragada and C. E. Brodley, "Class noise mitigation through instance weighting," in *European Conference on Machine Learning*, pp. 708–715, Springer, 2007.

[78] C. Bouveyron and S. Girard, "Robust supervised classification with mixture models: Learning from data with uncertain labels," *Pattern Recognition*, vol. 42, no. 11, pp. 2649–2658, 2009.

[79] N. El Gayar, F. Schwenker, and G. Palm, "A study of the robustness of knn classifiers trained using soft labels," in *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pp. 67–80, Springer, 2006.

[80] G. Shafer *et al.*, *A mathematical theory of evidence*, vol. 1. Princeton university press Princeton, 1976.

[81] P. Smets, "Decision making in the tbm: the necessity of the pignistic transformation," *International Journal of Approximate Reasoning*, vol. 38, no. 2, pp. 133–147, 2005.

[82] T. Denoeux, "A k-nearest neighbor classification rule based on dempster-shafer theory," *IEEE transactions on systems, man, and cybernetics*, vol. 25, no. 5, pp. 804–813, 1995.

[83] T. Denoeux, "A neural network classifier based on dempster-shafer theory," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 30, no. 2, pp. 131–150, 2000.

[84] A. Ganapathiraju, J. Picone, *et al.*, "Support vector machines for automatic data cleanup.," in *INTERSPEECH*, pp. 210–213, 2000.

[85] R. Rosales, G. Fung, and W. Tong, "Automatic discrimination of mislabeled training points for large margin classifiers," in *Proc. Snowbird Mach. Learn. Workshop*, pp. 1–2, 2009.

[86] O. Dekel and O. Shamir, "Good learners for evil teachers," in *Proceedings of the 26th annual international conference on machine learning*, pp. 233–240, ACM, 2009.

[87] C.-f. Lin *et al.*, "Training algorithms for fuzzy support vector machines with noisy data," *Pattern recognition letters*, vol. 25, no. 14, pp. 1647–1656, 2004.

[88] W. An and M. Liang, "Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises," *Neurocomputing*, vol. 110, pp. 101–110, 2013.

[89] R. Khardon and G. Wachman, "Noise tolerant variants of the perceptron algorithm," *Journal of Machine Learning Research*, vol. 8, no. Feb, pp. 227–248, 2007.

[90] A. Kowalczyk, A. J. Smola, and R. C. Williamson, "Kernel machines and boolean functions," in *Advances in Neural Information Processing Systems*, pp. 439–446, 2002.

[91] Y. Li and P. M. Long, "The relaxed online maximum margin algorithm," in *Advances in neural information processing systems*, pp. 498–504, 2000.

[92] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. New York, NY, USA: Cambridge University Press, 2000.

[93] W. Krauth and M. Mézard, "Learning algorithms with optimal stability in neural networks," *Journal of Physics A: Mathematical and General*, vol. 20, no. 11, p. L745, 1987.

[94] P. Clark and T. Niblett, "The cn2 induction algorithm," *Machine learning*, vol. 3, no. 4, pp. 261–283, 1989.

[95] C. Domingo and O. Watanabe, "Madaboost: A modification of adaboost," in *COLT*, pp. 180–189, 2000.

[96] N. C. Oza, "Boosting with averaged weight vectors," in *International Workshop on Multiple Classifier Systems*, pp. 15–24, Springer, 2003.

[97] N. C. Oza, "Aveboost2: Boosting for noisy data," in *International Workshop on Multiple Classifier Systems*, pp. 31–40, Springer, 2004.

[98] Y. Kim, "Averaged boosting: A noise-robust ensemble method," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 388–393, Springer, 2003.

[99] V. Gómez-Verdejo, M. Ortega-Moral, J. Arenas-García, and A. R. Figueiras-Vidal, "Boosting by weighting critical and erroneous samples," *Neurocomputing*, vol. 69, no. 7, pp. 679–685, 2006.

[100] A. Krieger, C. Long, and A. Wyner, "Boosting noisy data," in *ICML*, pp. 274–281, 2001.

[101] G. I. Webb, "Multiboosting: A technique for combining boosting and wagging," *Machine learning*, vol. 40, no. 2, pp. 159–196, 2000.

[102] I. Cantador and J. R. Dorronsoro, "Boosting parallel perceptrons for label noise reduction in classification problems," in *International Work-Conference on the Interplay Between Natural and Artificial Computation*, pp. 586–593, Springer, 2005.

[103] F. A. Breve, L. Zhao, and M. G. Quiles, "Semi-supervised learning from imperfect data through particle cooperation and competition," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pp. 1–8, IEEE, 2010.

[104] D. Guan, W. Yuan, Y.-K. Lee, and S. Lee, "Identifying mislabeled training data with the aid of unlabeled data," *Applied Intelligence*, vol. 35, no. 3, pp. 345–358, 2011.

[105] Y. Duan, Y. Gao, X. Ren, H. Che, and K. Yang, "Semi-supervised classification and noise detection," in *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on*, vol. 1, pp. 277–280, IEEE, 2009.

[106] M.-R. Amini and P. Gallinari, "Semi-supervised learning with explicit misclassification modeling," in *IJCAI*, vol. 3, pp. 555–560, Citeseer, 2003.

[107] M. R. Amini and P. Gallinari, "Semi-supervised learning with an imperfect supervisor," *Knowledge and Information Systems*, vol. 8, no. 4, pp. 385–413, 2005.

[108] A. Krithara, M. R. Amini, J.-M. Renders, and C. Goutte, "Semi-supervised document classification with a mislabeling error model," in *European Conference on Information Retrieval*, pp. 370–381, Springer, 2008.

[109] J. Zhang, V. S. Sheng, J. Wu, and X. Wu, "Multi-class ground truth inference in crowdsourcing with clustering.," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 1080–1085, 2016.

[110] A. McCallum, "Multi-label text classification with a mixture model trained by em," in *AAAI workshop on Text Learning*, pp. 1–7, 1999.

[111] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1297–1322, 2010.

[112] L. Aroyo and C. Welty, "The three sides of crowdtruth," *Journal of Human Computation*, vol. 1, pp. 31–34, 2014.

[113] M. Schaekermann, E. Law, A. C. Williams, and W. Callaghan, "Resolvable vs. irresolvable ambiguity: A new hybrid framework for dealing with uncertain ground truth," in *Proceedings of the 1st Workshop on Human-Centered Machine Learning at SIGCHI*, 2016.

[114] N.-C. Chen, M. Drouhard, R. Kocielnik, J. Suh, and C. R. Aragon, "Using machine learning to support qualitative coding in social science: Shifting the focus to ambiguity," *ACM Transactions on Interactive Intelligent Systems*, vol. 8, June 2018.

[115] Y. Yan, G. M. Fung, R. Rosales, and J. G. Dy, "Active learning from crowds," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 1161–1168, 2011.

[116] G. Krempl, D. Kottke, and M. Spiliopoulou, "Probabilistic active learning: Towards combining versatility, optimality and efficiency," in *International Conference on Discovery Science*, pp. 168–179, Springer, 2014.

[117] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognition*, vol. 45, no. 9, pp. 3084–3104, 2012.

[118] K. Brinker, J. Fürnkranz, and E. Hüllermeier, "A unified model for multilabel classification and ranking," in *Proceedings of the 2006 conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29–September 1, 2006, Riva del Garda, Italy*, pp. 489–493, IOS Press, 2006.

[119] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, 2006.

[120] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine learning*, vol. 39, no. 2-3, pp. 135–168, 2000.

[121] F. De Comité, R. Gilleron, and M. Tommasi, "Learning multi-label alternating decision trees from texts and data," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 35–49, Springer, 2003.

[122] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[123] A. Wieczorkowska, P. Synak, and Z. Raś, "Multi-label classification of emotions in music," *Intelligent Information Processing and Web Mining*, pp. 307–315, 2006.

[124] E. Spyromitros, G. Tsoumakas, and I. Vlahavas, "An empirical study of lazy multilabel classification algorithms," in *Hellenic conference on artificial intelligence*, pp. 401–406, Springer, 2008.

[125] W. Cheng and E. Hüllermeier, "Combining instance-based learning and logistic regression for multilabel classification," *Machine Learning*, vol. 76, no. 2, pp. 211–225, 2009.

[126] A. Clare and R. King, "Knowledge discovery in multi-label phenotype data," *Principles of data mining and knowledge discovery*, pp. 42–53, 2001.

[127] H. Blockeel, L. D. Raedt, and J. Ramon, "Top-down induction of clustering trees," in *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, (San Francisco, CA, USA), pp. 55–63, Morgan Kaufmann Publishers Inc., 1998.

[128] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances in neural information processing systems*, pp. 681–687, 2002.

[129] K. Crammer and Y. Singer, "A family of additive online algorithms for category ranking," *Journal of Machine Learning Research*, vol. 3, no. Feb, pp. 1025–1058, 2003.

[130] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning and Knowledge Discovery in Databases*, pp. 254–269, 2009.

[131] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," *Advances in knowledge discovery and data mining*, pp. 22–30, 2004.

[132] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," *Machine learning: ECML 2007*, pp. 406–417, 2007.

[133] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 995–1000, IEEE, 2008.

[134] J. Read, "A pruned problem transformation method for multi-label classification," in *Proc. 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008)*, vol. 143150, 2008.

[135] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels," in *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD08)*, pp. 30–44, 2008.

[136] J. Fürnkranz, "Round robin classification," *Journal of Machine Learning Research*, vol. 2, no. Mar, pp. 721–747, 2002.

[137] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, no. Aug, pp. 975–1005, 2004.

[138] S.-H. Park and J. Fürnkranz, "Efficient pairwise classification," *Machine Learning: ECML 2007*, pp. 658–665, 2007.

[139] E. L. Mencía, S.-H. Park, and J. Fürnkranz, "Efficient voting prediction for pairwise multilabel classification," *Neurocomputing*, vol. 73, no. 7, pp. 1164–1176, 2010.

[140] D. Kocev, "Ensembles for predicting structured outputs," *Informatica*, vol. 36, no. 1, 2012.

[141] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2013.

[142] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 614–622, ACM, 2008.

[143] E. Jamison and I. Gurevych, "Noise or additional information? leveraging crowdsource annotation item agreement for natural language tasks.," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 291–297, 2015.

[144] T. Goyal, T. McDonnell, M. Kutlu, T. Elsayed, and M. Lease, "Your behavior signals your reliability: Modeling crowd behavioral traces to ensure quality relevance annotations," in *Sixth AAAI Conference on Human Computation and Crowdsourcing*, 2018.

[145] B.-B. Gao, C. Xing, C.-W. Xie, J. Wu, and X. Geng, "Deep label distribution learning with label ambiguity," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2825–2838, 2017.

[146] M. Ling and X. Geng, "Soft video parsing by label distribution learning," *Frontiers of Computer Science*, vol. 13, no. 2, pp. 302–317, 2019.

[147] X. Jia, W. Li, J. Liu, and Y. Zhang, "Label distribution learning by exploiting label correlations," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[148] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data mining and knowledge discovery handbook*, pp. 667–685, Springer, 2009.

[149] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[150] T. M. Mitchell *et al.*, "Machine learning. wcb," 1997.

[151] L. Atlas, D. Cohn, R. Ladner, M. A. El-Sharkawi, and R. J. Marks, II, "Advances in neural information processing systems 2," ch. Training Connectionist Networks with Queries and Selective Sampling, pp. 566–573, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990.

[152] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine learning*, vol. 15, no. 2, pp. 201–221, 1994.

[153] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 150–157, The Morgan Kaufmann series in machine learning,(San Francisco, CA, USA), 1995.

[154] A. Fujii, T. Tokunaga, K. Inui, and H. Tanaka, "Selective sampling for example-based word sense disambiguation," *Computational Linguistics*, vol. 24, no. 4, pp. 573–597, 1998.

[155] H. Yu, "Svm selective sampling for ranking with application to data retrieval," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 354–363, ACM, 2005.

[156] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 3–12, Springer-Verlag New York, Inc., 1994.

[157] A. K. McCallumzy and K. Nigamy, "Employing em and pool-based active learning for text classification," in *Proc. International Conference on Machine Learning (ICML)*, pp. 359–367, Citeseer, 1998.

[158] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of machine learning research*, vol. 2, no. Nov, pp. 45–66, 2001.

[159] S. C. Hoi, R. Jin, and M. R. Lyu, "Large-scale text categorization by batch mode active learning," in *Proceedings of the 15th international conference on World Wide Web*, pp. 633–642, ACM, 2006.

[160] C. A. Thompson, M. E. Califf, and R. J. Mooney, "Active learning for natural language parsing and information extraction," in *ICML*, pp. 406–414, Citeseer, 1999.

[161] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the conference on empirical methods in natural language processing*, pp. 1070–1079, Association for Computational Linguistics, 2008.

[162] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proceedings of the ninth ACM international conference on Multimedia*, pp. 107–118, ACM, 2001.

[163] C. Zhang and T. Chen, "An active learning framework for content-based information retrieval," *IEEE transactions on multimedia*, vol. 4, no. 2, pp. 260–268, 2002.

[164] R. Yan, J. Yang, and A. Hauptmann, "Automatically labeling video data using multi-class active learning," in *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, (Washington, DC, USA), pp. 516–, IEEE Computer Society, 2003.

[165] A. G. Hauptmann, W.-H. Lin, R. Yan, J. Yang, and M.-Y. Chen, "Extreme video retrieval: Joint maximization of human and computer performance," in *Proceedings of the 14th ACM International Conference on Multimedia*, MM '06, (New York, NY, USA), pp. 385–394, ACM, 2006.

[166] G. Tur, D. Hakkani-Tür, and R. E. Schapire, "Combining active and semi-supervised learning for spoken language understanding," *Speech Communication*, vol. 45, no. 2, pp. 171–186, 2005.

[167] Y. Liu, "Active learning with support vector machine applied to gene expression data for cancer classification," *Journal of chemical information and computer sciences*, vol. 44, no. 6, pp. 1936–1941, 2004.

[168] D. Angluin, "Queries and concept learning," *Machine learning*, vol. 2, no. 4, pp. 319–342, 1988.

[169] D. Angluin, "Queries revisited," in *International Conference on Algorithmic Learning Theory*, pp. 12–31, Springer, 2001.

[170] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of artificial intelligence research*, vol. 4, no. 1, pp. 129–145, 1996.

[171] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proceedings of the eleventh international conference on machine learning*, pp. 148–156, 1994.

[172] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden markov models for information extraction," in *International Symposium on Intelligent Data Analysis*, pp. 309–318, Springer, 2001.

[173] C. E. Shannon, "A mathematical theory of communication," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, pp. 3–55, Jan. 2001.

[174] V. V. Fedorov, W. Studden, and E. Klimko, eds., *Theory of optimal experiments*. Probability and mathematical statistics, New York: Academic Press, 1972.

[175] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, (New York, NY, USA), pp. 287–294, ACM, 1992.

[176] P. Melville, S. M. Yang, M. Saar-Tsechansky, and R. Mooney, "Active learning for probability estimation using jensen-shannon divergence," in *European Conference on Machine Learning*, pp. 268–279, Springer, 2005.

[177] R. Burbidge, J. J. Rowland, and R. D. King, *Active Learning for Regression Based on Query by Committee*, pp. 209–218. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[178] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," in *Advances in neural information processing systems*, pp. 1289–1296, 2008.

[179] N. Roy and A. McCallum, "Toward optimal active learning through monte carlo estimation of error reduction," *ICML, Williamstown*, pp. 441–448, 2001.

[180] D. A. Cohn, "Neural network exploration using optimal experiment design," *Advances in neural information processing systems*, pp. 679–679, 1994.

[181] B. Settles, *Curious machines: Active learning with structured instances*. ProQuest, 2008.

[182] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *Proceedings of the twenty-first international conference on Machine learning*, p. 79, ACM, 2004.

[183] Z. Xu, R. Akella, and Y. Zhang, "Incorporating diversity and density in active learning for relevance feedback," in *European Conference on Information Retrieval*, pp. 246–257, Springer, 2007.

[184] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 1003–1011, Association for Computational Linguistics, 2009.

[185] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka, Jr., and T. M. Mitchell, "Coupled semi-supervised learning for information extraction," in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, (New York, NY, USA), pp. 101–110, ACM, 2010.

[186] J. Deng, J. Krause, and L. Fei-Fei, "Fine-grained crowdsourcing for fine-grained recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2013.

[187] D. Gurari, S. Jain, K. Grauman, and M. Betke, "Pull the plug? predicting if computers or humans should segment images," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 382–391, 2016.

[188] L. Biewald, *Why human-in-the-loop computing is the future of machine learning*, 2015 (accessed February 11, 2017). `http://www.computerworld.com/article/3004013/robotics/why-human-in-the-loop-computing-is-the-future-of-machine-learning.html`.

[189] M. O'Brien, *Google, Tesla, others wait for DMV's self-driving rules*, 2015 (accessed February 11, 2017). `http://www.mercurynews.com/2015/10/26/google-tesla-others-wait-for-dmvs-self-driving-rules/`.

[190] V. Ambati, S. Vogel, and J. G. Carbonell, "Active learning and crowd-sourcing for machine translation," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pp. 2169–2174, 2010.

[191] J. J. Morgan, "Human in the loop machine translation of medical terminology," tech. rep., DTIC Document, 2010.

[192] O. F. Zaidan and C. Callison-Burch, "Feasibility of human-in-the-loop minimum error rate training," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pp. 52–61, Association for Computational Linguistics, 2009.

[193] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[194] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A c-lstm neural network for text classification," *arXiv preprint arXiv:1511.08630*, 2015.

[195] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," *arXiv preprint arXiv:1603.03827*, 2016.

[196] A. Rao and N. Spasojevic, "Actionable and political text classification using word embeddings and lstm," *arXiv preprint arXiv:1607.02501*, 2016.

[197] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification.," in *AAAI*, vol. 333, pp. 2267–2273, 2015.

[198] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 649–657, Curran Associates, Inc., 2015.

[199] A. Sadilek, H. A. Kautz, and V. Silenzio, "Modeling spread of disease from social interactions," in *In Sixth AAAI International Conference on Weblogs and Social Media (ICWSM)*, pp. 322–329, 2012.

[200] M. J. Paul and M. Dredze, "You are what you tweet: Analyzing twitter for public health.," *ICWSM*, vol. 20, pp. 265–272, 2011.

[201] A. Tamersoy, M. De Choudhury, and D. H. Chau, "Characterizing smoking and drinking abstinence from social media," in *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, pp. 139–148, ACM, 2015.

[202] N. Schrading, C. O. Alm, R. Ptucha, and C. Homan, "#whyistayed,#whyileft: Microblogging to make sense of domestic abuse," in *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics, Denver, CO, USA*, pp. 1281–1286, 2015.

[203] M. J. Paul and M. Dredze, "Discovering health topics in social media using topic models," *PloS one*, vol. 9, no. 8, p. e103408, 2014.

[204] M. De Choudhury, S. Counts, and E. Horvitz, "Major life changes and behavioral markers in social media: case of childbirth," in *Computer Supported Cooperative Work, CSCW 2013, San Antonio, TX, USA, February 23-27, 2013*, pp. 1431–1442, 2013.

[205] M. De Choudhury, S. Counts, and E. Horvitz, "Predicting postpartum changes in emotion and behavior via social media," in *2013 ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '13, Paris, France, April 27 - May 2, 2013*, pp. 3267–3276, 2013.

[206] M. De Choudhury, M. Gamon, S. Counts, and E. Horvitz, "Predicting depression via social media.," in *ICWSM*, p. 2, 2013.

[207] M. Kumar, M. Dredze, G. Coppersmith, and M. De Choudhury, "Detecting changes in suicide content manifested in social media following celebrity suicides," in *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, HT '15, (New York, NY, USA), pp. 85–94, ACM, 2015.

[208] M. J. Brzozowski, "Watercooler: exploring an organization through enterprise social media," in *Proceedings of the ACM 2009 international conference on Supporting group work*, pp. 219–228, ACM, 2009.

[209] J. DiMicco, D. R. Millen, W. Geyer, C. Dugan, B. Brownholtz, and M. Muller, "Motivations for social networking at work," in *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pp. 711–720, ACM, 2008.

[210] M. Smith, D. L. Hansen, and E. Gleave, "Analyzing enterprise social media networks," in *Computational Science and Engineering, 2009. CSE'09. International Conference on*, vol. 4, pp. 705–710, IEEE, 2009.

[211] M. De Choudhury and S. Counts, "Understanding affect in the workplace via social media," in *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, (New York, NY, USA), pp. 303–316, ACM, 2013.

[212] M. De Choudhury, E. Kiciman, M. Dredze, G. Coppersmith, and M. Kumar, "Discovering shifts to suicidal ideation from mental health content in social media," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 2098–2110, ACM, 2016.

[213] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[214] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[215] F. A. Gers and E. Schmidhuber, "Lstm recurrent networks learn simple context-free and context-sensitive languages," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1333–1340, 2001.

[216] M. Heron and B. Tejada-Vera, "Deaths: leading causes for 2005.," *National Vital Statistics Reports: From the Centers for Disease Control and Prevention, National Center for Health Statistics, National Vital Statistics System*, vol. 58, no. 8, pp. 1–97, 2009.

[217] M. K. Nock, G. Borges, E. J. Bromet, C. B. Cha, R. C. Kessler, and S. Lee, "Suicide and suicidal behavior," *Epidemiologic Reviews*, vol. 30, no. 1, pp. 133–154, 2008.

[218] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[219] P. Burnap, W. Colombo, and J. Scourfield, "Machine classification and analysis of suicide-related communication on twitter," in *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, HT '15, (New York, NY, USA), pp. 75–84, ACM, 2015.

[220] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[221] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[222] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and A. N. Smith, "Improved part-of-speech tagging for online conversational text with word clusters," in *Proceedings of*

the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 380–390, Association for Computational Linguistics, 2013.

[223] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*. O'Reilly Media, Inc., 2009.

[224] J. L. Fleiss, "Measuring nominal scale agreement among many raters.," *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[225] K. Krippendorff, *Content analysis: An introduction to its methodology*. Sage, 2004.

[226] J. Geertzen, "Inter-Rater Agreement with Multiple Raters and Variables," 2016. [Online; accessed 17-February-2016].

[227] K. Evanini, D. Higgins, and K. Zechner, "Using amazon mechanical turk for transcription of non-native speech," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 53–56, Association for Computational Linguistics, 2010.

[228] M. C. Hughes and E. Sudderth, "Memoized online variational inference for dirichlet process mixture models," in *Advances in Neural Information Processing Systems*, pp. 1133–1141, 2013.

[229] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[230] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[231] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[232] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[233] T. Vieira, "Kl-divergence as an objective function." `https://timvieira.github.io/blog/post/2014/10/06/kl-divergence-as-an-objective-function/`, 2014. Online, Accessed August 14, 2019.

[234] "Mediaeval benchmarking initiative for multimedia evaluation," 2019. [Online; accessed 24-December-2019].

[235] "Quality control code for estimating the quality of the workers in crowdsourcing environments," 2019. [Online; accessed 24-December-2019].

[236] "Fact evaluation judgment dataset," 2013. [Online; accessed 24-December-2019].

[237] "Sentiment analysis judgment dataset," 2013. [Online; accessed 24-December-2019].

[238] Twitter, "Twitter Developer Terms." `https://developer.twitter.com/en/developer-terms/agreement-and-policy`, 2018. Online, Accessed June 5, 2019.

[239] Bureau of Labor Statistics, "Time use on an average work day for employed persons ages 25 to 54 with children," 2013.

[240] Bureau of Labor Statistics, "Occupational suicides – census of fatal occupational injuries," 2009.

[241] Hazards Magazine, "Work suicide," 2014.

[242] A. Archambault and J. Grudin, "A Longitudinal Study of Facebook, LinkedIn, & Twitter Use," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2741–2750, ACM, 2012.

[243] W. B. Schaufeli and A. B. Bakker, "Job Demands, Job Resources, and Their Relationship with Burnout and Engagement: A Multi-Sample Study," *Journal of Organizational Behavior*, vol. 25, no. 3, pp. 293–315, 2004.

[244] FE, "Figure Eight." `https://www.figure-eight.com/`, 2019. (Accessed June 5, 2019).

[245] MT, "Amazon Mechanical Turk." `https://www.mturk.com/`, 2019. (Accessed June 5, 2019).

[246] Y. Sakamoto, M. Ishiguro, and G. Kitagawa, "Akaike information criterion statistics," *Dordrecht, The Netherlands: D. Reidel*, vol. 81, 1986.

[247] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[248] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley framenet project," in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pp. 86–90, Association for Computational Linguistics, 1998.

[249] B. K. Kromer and D. J. Howard, *Comparison of ACS and CPS Data on Employment Status*, 2011 (accessed February 26, 2017). `https://www.census.gov/library/working-papers/2011/demo/SEHSD-WP2011-31.html`.

[250] E. L. Groshen, *The Employment Situation*, 2015 (accessed February 26, 2017). `https://www.census.gov/newsroom/cspan/2015/employment.html`.

[251] U.S. Bureau of Labor Statistics, "Employment situation," 2017. `https://www.bls.gov/news.release/empsit.toc.htm`.

[252] C. J. Fillmore, "Frame semantics and the nature of language," *Annals of the New York Academy of Sciences*, vol. 280, no. 1, pp. 20–32, 1976.

[253] C. J. Fillmore, "The case for case reopened," *Syntax and semantics*, vol. 8, no. 1977, pp. 59–82, 1977.

[254] C. Fillmore, "Frame semantics," *Linguistics in the morning calm*, pp. 111–137, 1982.

[255] C. J. Fillmore and C. F. Baker, "Frame semantics for text understanding," in *Proceedings of WordNet and Other Lexical Resources Workshop, NAACL*, 2001.

[256] J. Aguilar, C. Beller, P. McNamee, B. Van Durme, S. Strassel, Z. Song, and J. Ellis, "A comparison of the events and relations across ace, ere, tac-kbp, and framenet annotation standards," in *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pp. 45–53, 2014.

[257] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," *Computational linguistics*, vol. 28, no. 3, pp. 245–288, 2002.

Appendices

Appendix A

Supplementary materials for Section 4.3.1

In this thesis, we address the categorization problem of career changes and transitions expressed in job-related tweets, to illustrate our continuous work to build the humans-in-the-loop learning framework. We are interested in modeling and detecting the major employment stages in public social media. We aim to deepen our explorations of job-themed discourse and extract more fine-grained knowledge and associated behavior patterns (such as going for an interview, landing a new job, or getting laid off) from public social media data.

We first refer to the official census data about employment and unemployment, to understand how the government or official institutions define various employment stages. This provides us with regular rules to form this multi-class classification problem. Then we match a series of job/employment frames in FrameNet [248] to the official definitions of employment stages, to use the existing state-of-the-art frame-semantic parsing system as an auxiliary tool in our proposed humans-in-the-loop framework.

A.1 Employment Situations in Census Data

In published employment status surveys and reports, the U.S. Census Bureau defines the employment-classification concept regarding three categories that characterize and determine the individual relationship to the labor market: (1) **Employed**, (2) **Unemployed**, or (3) **Not in labor force**.

Each respondent in their surveys is classified in one and only one category (usually during a particular survey week window) [249, 250, 251]. The U.S. Bureau of Labor Statistics gives the definitions of the three types respectively: Employment[1], Unemployed[2], and Not in labor force[3]. When classifying the employment status for each tweet that was posted at a specific timestamp, it should be classified into only one category. Besides the above three categories, we will add an additional category (4) **Others** when none of the given categories is matched.

A.1.1 Employed

The **employed** are officially defined as:

- If people worked for pay (salary workers) or profit (self-employed) during the census survey reference week.
- If people worked in a family-operate business or farm (at least 15 hours per week without pay).
- If people were temporarily absent from their regular jobs (no matter they were paid or not during the time off) because of vocation, illness, maternity/paternity leave, family/personal obligations, labor dispute, bad weather or other short-term reasons.

A.1.2 Unemployed

People are classified as **unemployed** if:

- they do not have a job at all during the survey reference week.
- they made specific efforts to look for a job in the prior 4 weeks, such as: contacting an employer or employment agency, submitting resumes or job applications, placing or answering job advertisements, etc.
- they were available for work, such as expecting to be recalled from temporary layoff (unless temporarily ill).

[1] `https://www.bls.gov/cps/cps\_htgm.htm\#employed`
[2] `https://www.bls.gov/cps/cps\_htgm.htm\#unemployed`
[3] `https://www.bls.gov/cps/cps\_htgm.htm\#nilf`

A.1.3 Not In Labor Force

The labor force is defined to be made up of the employed and the unemployed. People are classified as **not in the labor force** if they both have no job and are not looking for one, for example:

- students who are going to school;
- seniors who are retired.

There are cases that people are not in the labor force, but *marginally attached to the labor force*, if:

- they indicated their desires to have a job
- they have looked for jobs in the last 12 months
- they were discouraged by job seeking process, but available for work

A.2 Mapping Frames to Employment Stages

FrameNet derived from the linguistic and lexicographic theory of Frame Semantics [252,253,254,255] with basic and straightforward ideas to achieve the goal of capturing information about events and relations in text. **Semantic frame** was universally introduced and represented as a type of descriptions of events, associated with additional information such as event participants (*frame elements*), relations between one event type to another (*frame relations*), and words/phrases to trigger a given frame (*lexical units*). Compared to classical ACE (Automatic Content Extraction) and ECE (Entities, Relations, Events) standards to annotate entities, events, and relations in a variety of documents, frames are much more comprehensive and finer grained [256]. FrameNet defines the frames, annotates sentences/documents, and demonstrates frame elements, frame relations and lexical units in syntactic structures.

Baker et al. [248] developed the Berkeley FrameNet Project and contributed an ongoing upgraded database of distinct semantic frames and corresponding frame-evoking lexical units. The newly released FrameNet (version 1.7) includes 1,222 identified frames and 13,586 lexical units. This system of frame representations provides us a lexical basis for further events and actions reasoning.

FrameNet has been heavily used as a lexical resource for Semantic Role Labeling (SRL) tasks [257]. We will start from FrameNet [248] as a reference of ontologies of job-related frame narratives, and build our annotation scheme of job-changing events in virtue of frames.

Frame-semantic parsing that returns frames and frame elements could play an important role in our humans-in-the-loop framework as it introduces semantic meanings when selecting samples for human annotators to examine. Before using frames directly as some linguistic features in our proposed multi-class classification problem, it is essential to map frames to the official employment stages as discussed in the previous section.

A.2.1 Job/Employment Frames

In FrameNet, there are some semantic frame types related to job and employment. They are detailed with their definitions[4] as:

1. Being_employed

   "*An Employee has a Position doing work in a particular Field, or doing work on a particular Task, for which an Employer gives Compensation to the Employee.*"

2. Employee_scenario

   "*The sequence of events in which the Employee hires on with an Employer, holds a Position, and finally leaves the Position.*"

3. Employer_scenario

   "*The sequence of events in which the Employer hires an Employee, employs them in a Position for some Duration, and finally lets them go from the Position.*"

4. Employing

   "*An Employer employs an Employee whose Position entails that the Employee perform certain Tasks in exchange for Compensation.*"

5. Employment_continue

[4] Adopted from `https://framenet.icsi.berkeley.edu/fndrupal/frameIndex`

"*This is a non-perspectivized frame representing the middle stage of the Employment_scenario, in which there is a stable employment relationship between the Employee and the Employer.*"

6. Employment_end

   "*This is a non-perspectivized frame representing the final stage of the Employment_scenario, in which the relationship between the Employer and the Employee comes to an end. There are two different ways in which the relation ends, represented in the Firing frame (in which the Employer is agentive) and the Quitting frame (in which the Employee is agentive).*"

7. Employment_scenario

   "*An Employee and Employer enter into an employment relation, wherein the Employee remains employed for some Duration of time, and finally the relationship ends either by the Employee leaving the job or the Employer letting go (or firing) the Employee. To each of these events there are concomitants, such as agreeing to/signing a contract for entering employment, compensation and performance of a service for the employment period itself, and severance for the dissolution of the relationship. There are several other events involved, including preparatory actions on the part of the Employer (posting the Position), the prospective Employee's part (looking for a job), or both (job interviews). In addition, there is the possibility of change in the relationship of Employer and Employee during the employment period, such as a change in Position (e.g. promotion, demotion) and a change in Compensation (e.g. raise, paycut).*"

8. Employment_start

   "*This is a non-perspectivized frame representing the initial stage of the Employment_scenario: the formation of the employment relationship between the Employer and the Employee.*"

9. Hiring

   "*An Employer hires an Employee, promising the Employee a certain Compensation in exchange for the performance of a job. The job may be described either in terms of a Task or a Position.*"

10. Quitting

    "*An Employee voluntarily leaves the service of an Employer.*"

11. Get_a_job

> "*A new Employee obtains a Position with an Employer, with which there are certain Tasks associated; in exchange for the performance of these Tasks, the Employee receives Compensation from the Employer.*"

These frames provides ways to answer the question in Section 4.3.1 about employment conditions revealed in job-related tweets. If further putting some of these frames in sequence, they could roughly form a job status cycle as illustrated in Figure 4.2, for example: Employment_start → Employment_continue → Employment_end → Get_a_job (→ Employment_start).